IDENTIFICATION AND DESIGN OF CONTROL SYSTEMS

Ph. D. THESIS

CHENG-KUNG YU

NOVEMBER 1985

IDENTIFICATION AND DESIGN OF CONTROL SYSTEMS

THESIS SUBMITTED TO THE DEPARTMENT OF ENGINEERING, UNIVERSITY OF WARWICK

for the degree
Ph.D IN ENGINEERING

by
CHENG-KUNG YU

NOVEMBER 1985

ACKNOWLEDGEMENT

The Author would like to express his deepest gratitude to Dr. P.K. Sinha for his guidance, encouragement and patience during the period of this research work.

The author is also grateful to professor J.L. Douce for his encouragement.

Acknowledgement is also due to Dr. Shih for his assistance in the design and construction of the deadbeat control hardware.

STATEMENT OF ORIGINALITY

This thesis is the work of the author and contains the following original contributions to the fields of control theory and control engineering :

1. A new analytical theory and an algorithm to identify the parameters of multi-input multi-output known order linear systems.
2. Formulation of a novel method to identify the system order and the unknown parameters of multi-input multi-output linear continuous systems.
3. A new algorithm to identify the system parameters and the order of linear discrete systems.
4. A new method of designing microprocessor-based deadbeat controller for digital servo systems with finite settling time.
5. A new method of designing digital systems using a parameter optimization method via non-linear programming and the finite-settling-time criterion.

IDENTIFICATION AND DESIGN OF CONTROL SYSTEM

SYNOPSIS

An extensive literature on parameter identification and design of multi-input multi-output control systems exists. Despite this and the presence of a wide ranging software in these areas of control engineering, there is an absence of analytical work and algorithms which are able to treat continuous linear and non-linear, as well as discrete systems within the same mathematical framework. Furthermore, virtually all currently available identification and design algorithms require large processing power/working storage and, consequently are suitable only to users with access to mainframe computers. This thesis is concerned with the development of a number of novel mathematical theories for parameter identification and design of linear and non-linear systems. These theories and their associated results are used to develop numerical algorithms suitable for implementation on low-cost personal computers with memory sizes of around 256 kbytes. The programs described in the thesis have been written by using IBM-4331 Fortran and can be run with little modification on IBM-PC Although there are constraints on accuracy, the analytical results and the associated

software developed in this thesis would be of use to practising control engineers in the prelimirary analysis/ design of physical systems.

The thesis is divided into two parts: Part I (Chapters 1 — 5) considers the problems of parameter identification of continuous linear time-invariant as well as non-linear and time-varying and discrete systems. Part II (Chapters 6 and 7) considers the problems of designing digital servo systems. The main contributions of this thesis are:

1. Developement of a new algorithm for the identification of the parameters of multi-input multi-output known order linear systems. This method is based on the integration of the completely controllable dynamical equation (Chapter 2).

2. Formulation of a novel method to identify the system order and the unknown parameters of multiple-input multiple-output linear time-invariant continuous systems. This method is based on the special structure of the system matrix and multiple integration of the dynamical equation [$\dot{\underline{x}}(t) = A\underline{x}(t) + B\,\underline{u}(t)$ and $\underline{y}(t) = C\,\underline{x}(t)$] (Chapter 3). In Chapter 4, these results are extended to time-varying and non-linear systems.

3. Identification of system parameters and order for linear discrete single-input single-output systems using the rank difference between two matrices constructed from the input-output measurement (Chapter 5). The procedure used to identify unknown parameters is based on a special output data vector $\underline{y}$ (K ; N) which is a linear combination of the input -output data The originality of the results presented in Chapters 2 — 5 lies in the fact that linear and non-linear as well as continuous and discrete systems may be identified by using a single mathematical framework.

The main advantage of this is that a self contained system package can be developed. Various parts of such a system software have been described in the thesis.

4. A microprocessor-based output deadbeat controller for digital servo systems with finite settling time is presented. The controller configuration introduced in this chapter can also be applied to implement the widely used digital lead-lag compensators and PID controllers (Chapter 6).

5. A new method of designing digital systems using the parameter optimization method via non-linear programming and the finite-settling time criterion (Chapter 7).

TABLE OF CONTENTS

SYNOPSIS

CHAPTER

PAGE

PART I .... a— 1

1 IDENTIFICATION TECHNIQUES .... 1— 1

2 KNOWN ORDER MIMO SYSTEMS .... 2— 1

2 — 1 Identification method .... 2— 2

2 — 2 Identification algorithm and illustrative examples .... 2—12

2 — 3 Concluding remarks .... 2—45

3 UNKNOWN ORDER MIMO SYSTEMS .... 3— 1

3 — 1 Identification from input-output data .... 3— 7

3 — 2 Order determination procedure .... 3—21

3 — 3 Parameter determination .... 3—42

3 — 4 Identification algorithm and illustrative examples .... 3—50

3 — 5 Concluding remarks .... 3—68

4 EXTENSION TO TIME-VARYING AND NON-LINEAR SYSTEMS .... 4— 1

4 — 1 Time - varying systems .... 4— 2

4 — 2 Non-linear systems .... 4— 7

4 — 3 Concluding remarks .... 4—10

5 DISCRETE SISO SYSTEMS ······························· 5— 1

5—1 Order determination and flow chart ··············· 5— 1

5—2 Numerical examples of order determination··· 5—13

5—3 Parameters determination and flow chart ······ 5—16

5—4 Numerical examples of parameters determination ·· 5—25

5—5 Concluding remarks ····································· 5—30

PART Ⅱ·· b

6 A MICROPROCESSOR-BASED OUTPUT DEADBEAT CONTROLLER ·· 6— 1

6—1 Deadbeat output compensation algorithm ········ 6— 2

6—2 Hardware system and implementation ············ 6— 6

6—3 Lead-lag compensator and PID controller ······ 6— 9

6—4 Concluding remarks ····································· 6—13

7 DESIGN OF DIGITAL SYSTEMS ······················· 7— 1

7—1 Digital control systems ······························ 7— 2

7—2 Performance index PI ································· 7— 5

7—3 Computational techiques ······························ 7— 6

7—4 Design example ··· 7—10

7—5 Condluding remarks ····································· 7—20

8 CONCLUDING COMMENT ································· 8— 1

REFERENCES ·· R— 1

APPENDIX 1 ··· A— 1

APPENDIX 2 ··· A— 9

APPENDIX 3 ··· A—23

APPENDIX 4 ··· A—31

PART- I

This part of the thesis is concerned with the identification of linear and non-linear continuous systems and linear discrete-time systems. The primary objective of this part is to present a comprehensive theory of identification with a view to developing a set of software suitable for small computers (typically the IBM-PC). The programs developed here are considered to be very cost effective in cases where a certain amount of inaccuracy may be acceptable as a compromise to obtain a fast, simple and easy to use identification technique through real-time measurement data. One of the main constraints of using the algorithm developed here is that the measurement data is assumed to be noise free. Since a substantial amount of work with noisy data has been undertaken in the past, this problem is not considered here.

This part of the thesis has the following chapters :

Chapter-1 Identification techniques

Chapter-2 Known order MIMO (multi-input multi-output) systems

Chapter-3 Unknown order MIMO systems

Chapter-4 Extension to time-varying and non-linear systems

Chapter-5 Discrete SISO (single-input single output) systems

Chapter - 1 IDENTIFICATION TECHNIQUES

System identification is generally accepted to be one of the most important problems in system analysis, synthesis, optimization and adaptation. In most identification algorithms, the structure of the system is assumed to be known, and the system is described by a proper mathematical model with unknown parameters. So the identification problem is reduced to a parameter estimation problem. Once the system is described by a mathematical model in terms of differential equations, state space equations or transfer functions, then the characteristic parameters are determined from experimental measurements of the system inputs, states and outputs. In some cases the system states are accessible for measurement; in many engineering systems some or all the states are not accessible for direct measurement, consequently only the system inputs and outputs should involved in the derivation of experimental identification algorithms.

Many factors need to be considered in the identification procedure such as the proper choice of the mathematical model to describe the system, assessment of the stored energy (initial conditions), and whether some special test signals are needed as the input to the system to obtain measurement data for appropriate estimation.

Various identification schemes have been proposed in addition to the well-known methods such as the learning model [1], the quasi-linearization [2], the auxiliary models [3], and the regression analysis method [4]. Ho and Kalman [5] used the external description of the system displayed via the Markov parameters with an impulse input and zero initial conditions. Puri and Waygant [3] considered single-input single-output systems, where auxiliary lag networks and a test signal of an impulse or white noise were used. Gopinath [6] considered multiple-input multiple-output linear-systems where differentiation of the system inputs and outputs is required which introduces noise and numerical errors. Khatwani and Bujwa [7] used an exponential test signal to identify the parameters of single-input single-output linear systems. K. S. Narendra and S.S. Tripathi [8] used the model reference approach to design a parameter adaptive model reference system. The unknown parameters are determined when the system states are accessible for measurement. G. Ludres and K. S. Narendra [9] used the model reference technique when only the system inputs and outputs are accessible for measurement. T. C. Hsia and V. Vimolvanich [10], used an analog method which is based on the learning model concept to determine the parameters of linear and non-linear systems. K. S. Kumar and R. Sridnar [2] applied the quasi-linearization method to identify the parameters of linear and non-linear systems. An identification algorithm was presented by A. Sherif and M. Y. Wu [11], where all the states of multiple-input multiple-output linear systems are assumed to be accessible for measurement, and that the system is settable, i.e., all initial states can be set to a desired value. Also a step input was required as a test signal. The above algorithm was used by H. Abedi [12] to identify the parameters of an actual distillation tower under normal operational conditions and satisfactory results were obtained.

(a) SYSTEM ORDER

The order of a dynamic system is a key parameter in system identification. Given a sequence of measurements from the input and the output terminals of a system, a general objective is to construct a system based on these available data, such that it can approximately give rise to the same input-output response. With the a-priori knowledge of the system order, suitable structure of correct dimension is first selected and then the system parameters estimated. However, if such an a-priori knowledge is lacking, the order of the system must be determined in addition to estimating the system parameters.

Procedures for system identification of certain single-input single-output linear systems requiring no knowledge of the system order have also been developed in the literature [31—33]. However, the knowledge of the upper bound of the system order is usually required so that higher order models can be assumed for practical applications.

The problem of order determination has been studied previously [13—21]. Graupe, Krause and Moore [13] presented a procedure for estimating the AR and MA orders of a white-noise excited ARMA (autoregressive moving-average) process from noise-free output measurements. Aström and Eykhoff [14] formulated a probability test for the noise-free scalar linear systems with noisy input and output observations. By fitting the least-squares models of different orders directly, Astrom and Eykhoff estimated the order by employing an F-test on the reduction of the loss function. Chow [15] formed a null hypothesis test for the order of a pure moving-average process with white-noise input from noise-free output observations. Chow [16] also gave a procedure based on the correlation functions of the output for estimating the orders of an ARMA process with white-noise input from noisy output measurements. Woodside [17] proposed three order test procedures for noise-free scalar linear systems with input and output measurements being corrupted by noises of known characteristics. Unbehauen and Gohring [18] compared seven order test procedures for their performance qualities based on large number of investigated cases. Comparison of order test procedures has also been made by Van Den Boom and Van Den Enden [19].

(b) MULTIVARIABLE SYSTEMS

All procedures summarized above were developed for single-input, single-output linear, time-invariant, discrete-time systems. For multivariable systems, the main difficulty in system identification is the existence of a class of canonical forms rather than a unique canonical form [21—24]. In the literature, order determination procedures for multivariable systems have been developed only for a class of discrete-time stationary processes. Mehra [20] presented a goodness-of-fit test procedure based on the innovation property [25] of an optimal Kalman filter [26 — 28] via the innovation approach [29 , 30] for the case where a time-invariant linear system is excited by white-noise. Tse and Weinert [21] dealt with the same white noise excited stationary process via a particular " canonical" construction. Starting from any output terminal, they investigate the increase in order due to additional output terminals based on a covariance matrix constructed from the noisy output observations.

(c) REDUCTION OF ORDER

In practice, even when it is possible to identify the parameters of complex and high order systems, the analysis, optimization and adaptation would require a large amount of computation. One way of overcoming these computational difficulties is to use a low order model of the high order system which is computationally and analytically more tractable than the actual system, yet provides sufficient information about the original system.

Different methods [34, 35, 36, 37, 38] have been proposed in the past for the reduction of high order transfer functions, or decreasing the dimension of the state equation.

In the conventional approach, Evans [37] obtained lower order systems by estimating the transients of a closed-loop transfer function from its dominant roots. Chen [39] and Biernson [40] used a different method, where the corner frequencies in a Bode diagram are ignored if the gains are below -15 db or above $+15$ db. Davison [36] considered the reduction of the dimension of the state equation by keeping the dominant eigenvalues and discarding the others.

The principle of the reduction techniques in the conventional approach is based on the use of the dominant roots. These methods are not always applicable, primarily because many control systems have no dominant roots.

Furthermore, the actual high order system parameters are required to be known, which is not always possible in practice.

Another reduction approach is to search, in some fashion, for the coefficients of a set of differential equations of specified order such that their response approaches that of the original system in such a manner that the mean square error between the two responses over a given finite interval is minimized. The methods proposed by Chidambara [41], Anderson [34] and Sinha and Pilli [38] belong to this category.

A different approach due to Chen and Shieh [35], is based on the principle of the continued fraction expansion of the transfer function of the actual high order system, which is required to be known.

N. K. Sinha and G. T. Bereznai [42] proposed a different technique. The technique is based on the pattern-search algorithm of Hook and Jeeves [43]. Starting from an approximate first or second-order model, an optimum model of that order is determined, and the process is continued with the order increasing progressively until the error criterion is satisfied or the desired order is reached. T. Yahagi [44] applied a parameter optimization technique to obtain low order models of high order systems by approximating the state equations. The error criterion function is given by the integral-squared error of the transient responses of the original system and its low order model.

(d) SCOPE OF THIS THESIS

In Part-I of this thesis, several identification procedures and associated algorithms are presented, these determine the system order (n) and the unknown parameters of a given system (A,B,C,D). Also a reduced order model of the given high order system can be determined by using one of these schemes.

In the Chapter 2, multiple-input multiple-output known order continuous systems are considered. The basic procedure is applicable to systems where all system states are accessible for measurement. All that is required in this scheme, is a set of integrators depending on the system order and the number of inputs and outputs. In this chapter a software package is developed using the analytical results of this scheme (Fig. 1-1) . The main advantage of this proposed method is that no special test signal is required, which makes it very suitable for on-line identification.

Chapter 3 considers the order determination and parameter determination of continuous-time linear systems. The procedure employed for system identification is based on the special structure of matrix (A) and the multiple-integration of dynamic equation. In this method no special test signal is required, which makes this method applicable for on-line identification.

Chapter 4 extends the above results to the unknown order multiple-input multiple-output nonlinear systems, and time-varying systems. This identification technique requires no knowledge of the system order or the initial conditions. Furthermore, in addition to the parameter determination, the initial conditions can be determined by this scheme.

The problem of identifying discrete linear system is considered in the last chapter in Part-I of the thesis. Chapter 5 develops a method of determining the system order and parameters. This method is considered with reference to single-input single-output systems primarily to highlight the applicability of the general identification philosophy developed in Chapters 2—4 to discrete systems. A topic of future research could be to extend these results to multi-input multi-output discrete system to complement the proceding development for continuous systems.

Part-II of the thesis has two chapters. Chapter 6 presents a microprocessor-based output deadbeat controller for digital servo system with finite settling time. The controller

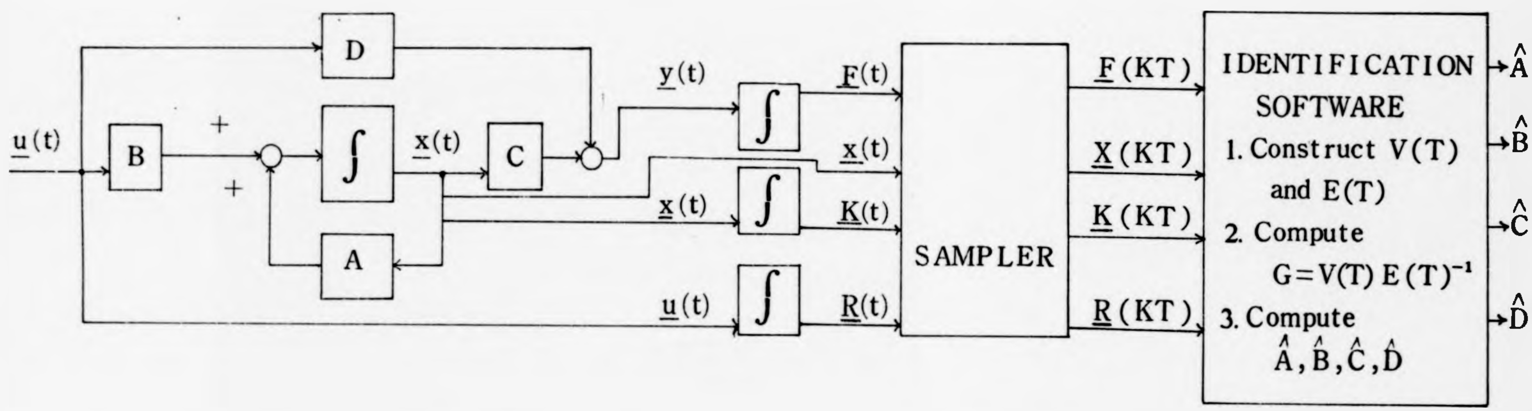

Fig. 1−1 The basic configuration of identification algorithm configuration introduced in this chapter can also be applied to implement the widely used digital lead-lag compensators and PID controllers.

In Chapter 7 a computer aided design technique for digital control systems with finite settling time is presented. A performance index (PI), consisting of three terms for penalizing the system error, the actuating force, and the location of the closed-loop poles , is formulated. The parameters of the digital controllers are determined by minimizing this performance index. A CAD algorithm is described through a numerical example to demonstrate the stages of using this design technique.

Chapter — 2 KNOWN ORDER MIMO SYSTEMS

This chapter develops a theoretical framework of estimating the parameters of a multi-input multi-output (MIMO) systems whose order is known a-priori. This problem has considerable practical significance, especially in cases where a somewhat approximate description is needed to assess the general behaviour of a system. The main advantage of this method is its simplicity— requiring only one matrix inversion and one matrix multiplication. The consequent disadvantage is that some inaccuracy in the estimated system parameters may have to be accepted if the system order is very high. Some, rather simple, numerical examples are included to study the extent of error introduced in the identified parameters.

The MIMO linear time-invariant systems considered here is described by S :

$$S : \dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t) \tag{2-1}$$

$$\underline{y}(t) = C\,\underline{x}(t) + D\,\underline{u}(t) \tag{2-2}$$

where

$\underline{x}(t) \in R^{n\times 1}$ is a state vector

$\underline{y}(t) \in R^{p\times 1}$ is a output vector

$\underline{u}(t) \in R^{m\times 1}$ is a input vector

and $A \in R^{n\times n}$, $B \in R^{n\times m}$, $C \in R^{p\times n}$ and $D \in R^{p\times m}$ are unknown real constant matrices to be determined from experimental data.

2 — 1 Identification method

Consider the system S defined by (2—1) and (2—2). We assume here that the n-state vector $\underline{x}(t)$, the p-output vector $\underline{y}(t)$ and the m-input vector $\underline{u}(t)$ are all accessible for measurement.

Define :

$$\begin{bmatrix} k_1(t) \\ k_2(t) \\ \vdots \\ k_n(t) \end{bmatrix} \triangleq \int_{t_0}^{t} \begin{bmatrix} x_1(\tau) \\ x_2(\tau) \\ \vdots \\ x_n(\tau) \end{bmatrix} d\tau$$

or

$$\underline{k}(t) \triangleq \int_{t_0}^{t} \underline{x}(\tau)\, d\tau \tag{2-3}$$

$$\begin{bmatrix} R_1(t) \\ R_2(t) \\ \vdots \\ R_m(t) \end{bmatrix} \triangleq \int_{t_0}^{t} \begin{bmatrix} u_1(\tau) \\ u_2(\tau) \\ \vdots \\ u_m(\tau) \end{bmatrix} d\tau$$

or

$$\underline{R}(t) \triangleq \int_{t_0}^{t} \underline{u}(\tau)\, d\tau \qquad (2\text{—}4)$$

$$\begin{bmatrix} F_1(t) \\ F_2(t) \\ \vdots \\ F_p(t) \end{bmatrix} \triangleq \int_{t_0}^{t} \begin{bmatrix} y_1(\tau) \\ y_2(\tau) \\ \vdots \\ y_p(\tau) \end{bmatrix} d\tau$$

or

$$\underline{F}(t) \triangleq \int_{t_0}^{t} \underline{y}(\tau)\, d\tau \qquad (2\text{—}5)$$

Now, if both sides of (2—1) and (2—2) are integrated, we get:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} - \begin{bmatrix} x_1(t_0) \\ x_2(t_0) \\ \vdots \\ x_n(t_0) \end{bmatrix} = A \begin{bmatrix} k_1(t) \\ k_2(t) \\ \vdots \\ k_n(t) \end{bmatrix} + B \begin{bmatrix} R_1(t) \\ R_2(t) \\ \vdots \\ R_m(t) \end{bmatrix} \qquad (2\text{—}6a)$$

and

$$\begin{bmatrix} F_1(t) \\ F_2(t) \\ \vdots \\ F_p(t) \end{bmatrix} = C \begin{bmatrix} k_1(t) \\ k_2(t) \\ \vdots \\ k_n(t) \end{bmatrix} + D \begin{bmatrix} R_1(t) \\ R_2(t) \\ \vdots \\ R_m(t) \end{bmatrix} \qquad (2\text{—}6b)$$

In compact form, equations (2—6a) and (2—6b) may be expressed as

$$\underline{x}(t) - \underline{x}(t_0) = A\,\underline{k}(t) + B\,\underline{R}(t) \qquad (2\text{—}7)$$

$$\underline{F}(t) = C\,\underline{k}(t) + D\,\underline{R}(t) \qquad (2\text{—}8)$$

Equations (2—7) and (2—8) can be put in a partitioned form as:

$$\left[\begin{array}{c}\underline{x}(t)-\underline{x}(t_0)\\ \hline \underline{F}(t)\end{array}\right]=\left[\begin{array}{c|c}A & B\\ \hline C & D\end{array}\right]\left[\begin{array}{c}\underline{k}(t)\\ \hline \underline{R}(t)\end{array}\right] \qquad (2\text{—}9)$$

Define :

$$\underline{V}(t)\triangleq\left[\begin{array}{c}\underline{x}(t)-\underline{x}(t_0)\\ \hline \underline{F}(t)\end{array}\right] \qquad (2\text{—}10)$$

$$\underline{E}(t)\triangleq\left[\begin{array}{c}\underline{k}(t)\\ \hline \underline{R}(t)\end{array}\right] \qquad (2\text{—}11)$$

and

$$G\triangleq\left[\begin{array}{c|c}A & B\\ \hline C & D\end{array}\right] \qquad (2\text{—}12)$$

where

$$\underline{V}(t)\in R^{(n+p)\times 1}\ ,\ \underline{E}(t)\in R^{(n+m)\times 1}$$

and

$G\in R^{(n+p)\times(n+m)}$ is constant matrix

Then from (2—9) we have :

$$\underline{V}(t)=G\,\underline{E}(t) \qquad (2\text{—}13)$$

Now, if all n-states, p-outputs, m-inputs and their corresponding integrals are measured at $(n+m)$ successive equal intervals of time, with a sampling period T, we have :

for $t=T$

$$\left[\begin{array}{c}x_1(T)-x_1(t_0)\\ x_2(T)-x_2(t_0)\\ \vdots\\ x_n(T)-x_n(t_0)\\ \hline F_1(T)\\ F_2(T)\\ \vdots\\ F_p(T)\end{array}\right]=G\left[\begin{array}{c}k_1(T)\\ k_2(T)\\ \vdots\\ k_n(T)\\ \hline R_1(T)\\ R_2(T)\\ \vdots\\ R_m(T)\end{array}\right]$$

or $\underline{V}(T) = G\,\underline{E}(T)$

for $t = 2T$

$$\begin{bmatrix} x_1(2T) - x_1(t_0) \\ x_2(2T) - x_2(t_0) \\ \vdots \\ x_n(2T) - x_n(t_0) \\ \hline F_1(2T) \\ F_2(2T) \\ \vdots \\ F_p(2T) \end{bmatrix} = G \begin{bmatrix} k_1(2T) \\ k_2(2T) \\ \vdots \\ k_n(2T) \\ \hline R_1(2T) \\ R_2(2T) \\ \vdots \\ R_m(2T) \end{bmatrix}$$

or $\underline{V}(2T) = G\,\underline{E}(2T)$

for $t = (n+m)T$

$$\begin{bmatrix} x_1[(n+m)T] - x_1(t_0) \\ x_2[(n+m)T] - x_2(t_0) \\ \vdots \\ x_n[(n+m)T] - x_n(t_0) \\ \hline F_1[(n+m)T] \\ F_2[(n+m)T] \\ \vdots \\ F_p[(n+m)T] \end{bmatrix} = G \begin{bmatrix} k_1[(n+m)T] \\ k_2[(n+m)T] \\ \vdots \\ k_n[(n+m)T] \\ \hline R_1[(n+m)T] \\ R_2[(n+m)T] \\ \vdots \\ R_m[(n+m)T] \end{bmatrix}$$

or $\underline{V}[(n+m)T] = G\,\underline{E}[(n+m)T]$

or, in the compact form

$$
\left[\begin{array}{c|c|c|c|c}
x_1(T)-x_1(t_0) & x_1(2T)-x_1(t_0) & & & x_1[(n+m)T]-x_1(t_0) \\
x_2(T)-x_2(t_0) & x_2(2T)-x_2(t_0) & & & x_2[(n+m)T]-x_2(t_0) \\
\vdots & \vdots & & & \vdots \\
x_n(T)-x_n(t_0) & x_n(2T)-x_n(t_0) & & & x_n[(n+m)T]-x_n(t_0) \\
\hline
F_1(T) & F_1(2T) & & & F_1[(n+m)T] \\
F_2(T) & F_2(2T) & & & F_2[(n+m)T] \\
\vdots & \vdots & & & \vdots \\
F_p(T) & F_p(2T) & & & F_p[(n+m)T]
\end{array}\right] = G \cdot
$$

$$
\left[\begin{array}{c|c|c|c|c}
k_1(T) & k_1(2T) & & & k_1[(n+m)T] \\
k_2(T) & k_2(2T) & & & k_2[(n+m)T] \\
\vdots & \vdots & & & \vdots \\
k_n(T) & k_n(2T) & \cdots\cdots & \cdots\cdots & k_n[(n+m)T] \\
\hline
R_1(T) & R_1(2T) & & & R_1[(n+m)T] \\
R_2(T) & R_2(2T) & & & R_2[(n+m)T] \\
\vdots & \vdots & & & \vdots \\
R_m(T) & R_m(2T) & & & R_m[(n+m)T]
\end{array}\right]
$$

and Define:

$$
V(T) \triangleq [\underline{V}(T)\underline{V}(2T)\cdots\cdots\underline{V}[(n+m)T]] \qquad (2\text{—}14)
$$

and

$$E(T) \triangleq [\,\underline{E}(T)\underline{E}(2T)\cdots\cdots\underline{E}[(n+m)T]\,] \tag{2-15}$$

where

$$V(T) \in R^{(n+p)\times(n+m)}$$

$$E(T) \in R^{(n+m)\times(n+m)}$$

Then from (2-13) we have

$$V(T) = G\,E(T) \tag{2-16}$$

As will be seen in the identification algorithm $E(T)$ is required to be non-singular for the determination of the system parameters. A necessary condition for $E(T)$ to be non-singular is given by the following Lemma :

Lemma 2-1 A necessary condition for the matrix $E(T)$ to be non-singular is that the system to be identified is completely controllable.

Proof :

Consider equation (2-1)

Define a new augmented state vector $\hat{\underline{x}}(t)$ where

$$\hat{\underline{x}}(t) \triangleq \begin{bmatrix} \underline{x}(t) \\ \hline \underline{u}(t) \end{bmatrix} \tag{2-17}$$

and

$$\hat{\underline{x}}(t) \in R^{J\times 1} \quad , \; J = n+m$$

then

$$\begin{bmatrix} \dot{\underline{x}}(t) \\ \hline \dot{\underline{u}}(t) \end{bmatrix} = \left[\begin{array}{c|c} A & B \\ \hline 0 & 0 \end{array}\right] \begin{bmatrix} \underline{x}(t) \\ \hline \underline{u}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \hline I \end{bmatrix} \dot{\underline{u}}(t)$$

In compact form , we have :

$$\dot{\hat{\underline{x}}}(t)=\hat{A}\,\hat{\underline{x}}(t)+\hat{B}\,\dot{\underline{u}}(t) \tag{2-18}$$

where

$$\hat{A} \triangleq \left[\begin{array}{c|c} A & B \\ \hline 0 & 0 \end{array}\right] \tag{2-19}$$

and

$$\hat{B} \triangleq \left[\begin{array}{c} O_{n\times n} \\ \hline I_{m\times m} \end{array}\right] \tag{2-20}$$

$$\hat{A} \in R^{J\times J} \text{ and } \hat{B} \in R^{J\times m}$$

Define:

$$\underline{E}(t) \triangleq \int_{t_0}^{t} \hat{\underline{x}}(\tau)\,d\tau$$

$$= \left[\begin{array}{c} \underline{k}(t) \\ \hline \underline{R}(t) \end{array}\right] \quad \text{and} \quad \underline{E}(t_0) = \underline{O}$$

Then $\underline{E}(t)$ is the same as defined by (2—11).

Integrating both sides of (2—18)

$$\int_{t_0}^{t} \dot{\hat{\underline{x}}}(\tau)\,d\tau = \int_{t_0}^{t} \hat{A}\,\hat{\underline{x}}(\tau)\,d\tau + \int_{t_0}^{t} \hat{B}\,\dot{\underline{u}}(\tau)\,d\tau$$

we get :

$$\hat{\underline{x}}(t) - \hat{\underline{x}}(t_0) = \hat{A}\,\underline{E}(t) + \hat{B}\,[\,\underline{u}(t) - \underline{u}(t_0)\,]$$

or

$$\hat{\underline{x}}(t) = \dot{\underline{E}}(t) = \hat{A}\,\underline{E}(t) + \hat{B}\,[\,\underline{u}(t) - \underline{u}(t_0)\,] + \hat{\underline{x}}(t_0) \tag{2-21}$$

From (2—20)

$$\hat{B}\,\underline{u}(t_0) = \left[\begin{array}{c} O \\ \hline I \end{array}\right] \underline{u}(t_0) = \left[\begin{array}{c} O \\ \hline \underline{u}(t_0) \end{array}\right]$$

therefore

$$\hat{\underline{x}}(t_0) - \hat{B}\,\underline{u}(t_0) = \begin{bmatrix} \underline{x}(t_0) \\ \hline \underline{u}(t_0) \end{bmatrix} - \begin{bmatrix} O \\ \hline \underline{u}(t_0) \end{bmatrix} = \begin{bmatrix} \underline{x}(t_0) \\ \hline O \end{bmatrix}$$

Define :

$$W = \begin{bmatrix} \underline{x}(t_0) \\ \hline O \end{bmatrix} \tag{2-22}$$

where $W \in R^J$ is a constant vector.

Define :

$\hat{\underline{u}}(t) \in R^m$ such that

$$\hat{B}\,\hat{\underline{u}}(t) = \hat{B}\,\underline{u}(t) + W \tag{2-23}$$

Then equation (2—21) becomes

$$\dot{\underline{E}}(t) = \hat{A}\,\underline{E}(t) + \hat{B}\,\hat{\underline{u}}(t) \tag{2-24}$$

The solution of equation (2—24) is given by :

$$\underline{E}(t) = \phi(t - t_0)\,\underline{E}(t_0) + \int_{t_0}^{t} \phi(t-\tau)\,\hat{B}\,\hat{\underline{u}}(\tau)\,d\tau \tag{2-25}$$

where $\phi(t - t_0) = e^{\hat{A}(t-t_0)}$ is the transition matrix.

The state transition matrix can be written as :

$$\begin{aligned} \phi(t-t_0) &= e^{\hat{A}(t-t_0)} \\ &= a_0(t-t_0)\hat{A}^0 + a_1(t-t_0)\hat{A} + a_2(t-t_0)\hat{A}^2 + \cdots\cdots \\ &= \underline{I} + a_1(t-t_0)\hat{A} + a_2(t-t_0)\hat{A}^2 + a_3(t-t_0)\hat{A}^3 + \cdots \\ &= \sum_{i=0}^{J-1} a_i(t-t_0)\hat{A}^i \end{aligned} \tag{2-26}$$

where $a_i(t)$ are scalar valued functions.

Let

$$\underline{\hat{u}}(t) = \sum_{j=1}^{m} \hat{u}_j(t)\underline{e}_j \tag{2-27}$$

where $\underline{e}_j \in R^m$ is a unit vector which has a 1 in the j th place and zeros elsewhere.

Since $\underline{E}(t_0) = \underline{0}$, then equation (2—25) can be written as :

$$\underline{E}(t) = \sum_{i=0}^{J-1} \sum_{j=1}^{m} \int_{t_0}^{t} a_i(t-\tau)\hat{A}^i \hat{B}_j \hat{u}_j(\tau)\,d\tau$$

where $\hat{B}_j$ denotes the jth column of the matrix $\hat{B}$.

Define :

$$r_{ij}(t) \triangleq \int_{t_0}^{t} a_i(t-\tau)\hat{u}_j(\tau)\,d\tau \qquad i=0,1,2,\cdots\cdots$$

$$\cdots\cdots, J-1 \text{ and } j=1,2,\cdots\cdots,m \tag{2-28}$$

then

$$\underline{E}(t) = \sum_{i=0}^{J-1} \sum_{j=1}^{m} r_{ij}(t)\hat{A}^i \hat{B}_j \tag{2-29}$$

In a matrix form (2—29) can be written as following

$$\underline{E}(t) = \hat{Q}\underline{h}(t) \tag{2-30}$$

where

$$\hat{Q} \triangleq [\,\hat{B} \quad \hat{A}\hat{B} \cdots\cdots \hat{A}^{J-1}\hat{B}\,] \tag{2-31}$$

$$\hat{Q} \in R^{J \times Jm}$$

and

$$\underline{h}(t) = [\, r_{01}(t) \quad r_{02}(t) \cdots\cdots r_{0m}(t) \mid r_{11}(t) \cdots\cdots r_{1m}(t) \mid$$
$$\cdots\cdots \mid \cdots\cdots \mid \cdots\cdots \mid r_{(J-1)1} \cdots\cdots r_{(J-1)m}\,]^T \tag{2-32}$$

$$\underline{h}(t) \in R^{Jm}$$

At (J) successive intervals of time, with a sampling period T, equation (2—30) can be written as :

$$E(T) = \hat{Q}H(T) \qquad (2\text{—}33)$$

where

$$E(T) \triangleq [\,\underline{E}(T)\underline{E}(2T)\cdots\cdots\underline{E}(JT)\,] \qquad (2\text{—}34)$$

and

$$H(T) \triangleq [\,\underline{h}(T)\underline{h}(2T)\cdots\cdots\underline{h}(JT)\,] \qquad (2\text{—}35)$$

$E(T) \in R^{J\times J}$ and $H(T) \in R^{Jm\times J}$

For $E(T)$ to be non-singular, it is necessary that the rank of $E(T)$ is equal to J

i.e., $\rho\,[\,E(T)\,] = J$

From equation (2—33) we have

$$\rho\,[\,E(T)\,] \leq \min\,[\,\rho(\hat{Q}),\ \rho\,[\,H(T)\,]\,] \qquad (2\text{—}36)$$

Consider equation (2—28). where $a_i(t)$ are scalar valued function. Now if the input $u_j(t)$ $j = 1, 2, \cdots\cdots, m$ are linearly independent, then $r_{ij}(t)$ are linearly dependent, this implies that :

$$\rho\,[\,H(T)\,] = J$$

Then from (2—36), for $E(T)$ to be non-singular, it is necessary that

$$\rho(\hat{Q}) = J$$

By definition, we have

$$\hat{B} = \left[\begin{array}{c} O \\ \hline I \end{array}\right] \quad \text{and} \quad \hat{A} = \left[\begin{array}{c|c} A & B \\ \hline O & O \end{array}\right]$$

Then

$$\hat{A}\hat{B} = \left[\begin{array}{c} B \\ \hline O \end{array}\right],\ \hat{A}^2\hat{B} = \left[\begin{array}{c} AB \\ \hline O \end{array}\right],\ \cdots\cdots,\ \hat{A}^i\hat{B} = \left[\begin{array}{c} A^{i-1}B \\ \hline O \end{array}\right]$$

$$\hat{Q} = \left[\begin{array}{c|c|c|c|c} O & B & AB & \cdots\cdots & A^{J-2}B \\ \hline I_{m\times m} & O & O & \cdots\cdots & O \end{array}\right] \qquad (2-37)$$

It is clear from (2—37) that the matrix $[\, I_{m\times m}\ 0\ 0 \cdots\cdots 0\,]$ has m-linearly independent rows.

Thus if the original system given by (2—1) is completely controllable then the matrix $[\,0\ B\ \ AB\cdots\cdots A^{J-2}B\,]$ has n-linearly independent rows.

Therefore $\rho(\hat{Q}) = J = n + m$ and hence $\rho\,[E(T)] = J$

So for some linearly independent input $u_j(t)$ $j = 1, 2, \cdots \cdots\cdots, m$, and a sampling interval T, a necessary condition for E(T) to be non-singular is that the system is completely controllable. This completes the proof.

2 — 2 Identification algorithm and illustrative examples

This section develops the identification algorithm and the flowchart of computer program using the theory introduced in the section 2 — 1.

The identification algorithm is as follows:

Step 1 Consider the multiple-input multiple-output known order linear time-invariant system depicted in equations (2 — 1) and (2 — 2)

$$\underline{\dot{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t) \qquad ((2-1))$$

$$\underline{y}(t)=C\,\underline{x}(t)+D\,\underline{u}(t) \qquad ((2\text{—}2))$$

Step 2 In order to simulate and identify it with the digital computer, we must transform the continuous model into the discrete model as :

$$\underline{x}[(k+1)T]=\phi(T)\underline{x}(kT)+\theta(T)\underline{u}(kT) \qquad (2\text{—}38)$$

$$\underline{y}(kT)=C\underline{x}(kT)+D\underline{u}(kT) \qquad (2\text{—}39)$$

where

$$\phi(T)=e^{AT} \qquad (2\text{—}40)$$

$$\theta(T)=A^{-1}[e^{AT}-I]B \qquad (2\text{—}41)$$

Step 3 Compute $\phi(T)=I+AT+\dfrac{(AT)^2}{2!}+\cdots\cdots$ and $\theta(T)$.

Step 4 Compute $\underline{x}(kT)$ from (2—38) using $\phi(T)$ and $\theta(T)$ for $k=1,2,\cdots\cdots,(n+m)$

Step 5 Compute $\underline{R}(kT)$ for $k=1,2,\cdots\cdots,(n+m)$

$$\underline{R}(kT)\triangleq\int_0^{kT}\underline{u}(\tau)\,d\tau$$

Step 6 From equation (2—8), we have

$$\underline{x}(kT)-\underline{x}(0)=A\underline{k}(kT)+B\underline{R}(kT) \qquad (2\text{—}42)$$

then

$$\underline{k}(kT)=A^{-1}[\underline{x}(kT)-\underline{x}(0)-B\underline{R}(kT)] \qquad (2\text{—}43)$$

where

$$\underline{k}(kT)\triangleq\int_0^{kT}\underline{x}(\tau)\,d\tau$$

Step 7 From equation (2—9), we have

$$\underline{F}(kT)=C\underline{k}(kT)+D\underline{R}(kT) \qquad (2\text{—}44)$$

where

$$\underline{F}(kT) \triangleq \int_0^{kT} \underline{y}(\tau)\, d\tau$$

Therefore, we can get $\underline{k}(kT)$ and $\underline{F}(kT)$, $k=1,2,\cdots\cdots,(n+m)$.

Until now, we have obtained the data $\underline{x}(kT)$, $\underline{R}(kT)$, $\underline{k}(kT)$ and $\underline{F}(kT)$ for identification.

Step 8 Use $\underline{x}(kT)$, $\underline{R}(kT)$, $\underline{k}(kT)$ and $\underline{F}(KT)$ to construct the matrices $V(T)$ and $E(T)$ respectively.

$$V(T) \triangleq [\,\underline{V}(T)\underline{V}(2T)\cdots\cdots\cdots\cdots\underline{V}(n+m)T\,]$$

$$= \begin{bmatrix} x_1(T)-x_1(0) & x_1(2T)-x_1(0)\cdots\cdots & x_1((n+m)T)-x_1(0) \\ \vdots & \vdots & \vdots \\ x_n(T)-x_n(0) & x_n(2T)-x_n(0)\cdots\cdots & x_n((n+m)T)-x_n(0) \\ \hline F_1(T) & F_1(2T) & F_1((n+m)T) \\ F_2(T) & F_2(2T) & F_2((n+m)T) \\ \vdots & \vdots & \vdots \\ F_p(T) & F_p(2T) & F_p((n+m)T) \end{bmatrix}$$

(2—45)

$$E(T) \triangleq [\,\underline{E}(T)\quad \underline{E}(2T)\cdots\cdots\underline{E}((n+m)T)\,]$$

$$= \begin{bmatrix} k_1(T) & k_1(2T)\cdots\cdots & k_1((n+m)T) \\ k_2(T) & k_2(2T)\cdots\cdots & k_2((n+m)T) \\ \vdots & \vdots & \vdots \\ k_n(T) & k_n(2T)\cdots\cdots & k_n((n+m)T) \\ \hline R_1(T) & R_1(2T)\cdots\cdots & R_1((n+m)T) \\ R_2(T) & R_2(2T)\cdots\cdots & R_2((n+m)T) \\ \vdots & \vdots & \vdots \\ R_m(T) & R_m(2T)\cdots\cdots & R_m((n+m)T) \end{bmatrix} \qquad (2\text{—}46)$$

Step 9 Compute the matrix G from (2—16)

$$G = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = V(T)\, E(T)^{-1} \tag{2—47}$$

Flow-chart of the computer program, based on the above algorithm, is shown in Fig. 2—1 and the listing of the program (Program A) is given in Appendix 1, the numerical method is illustrated through a number of examples.

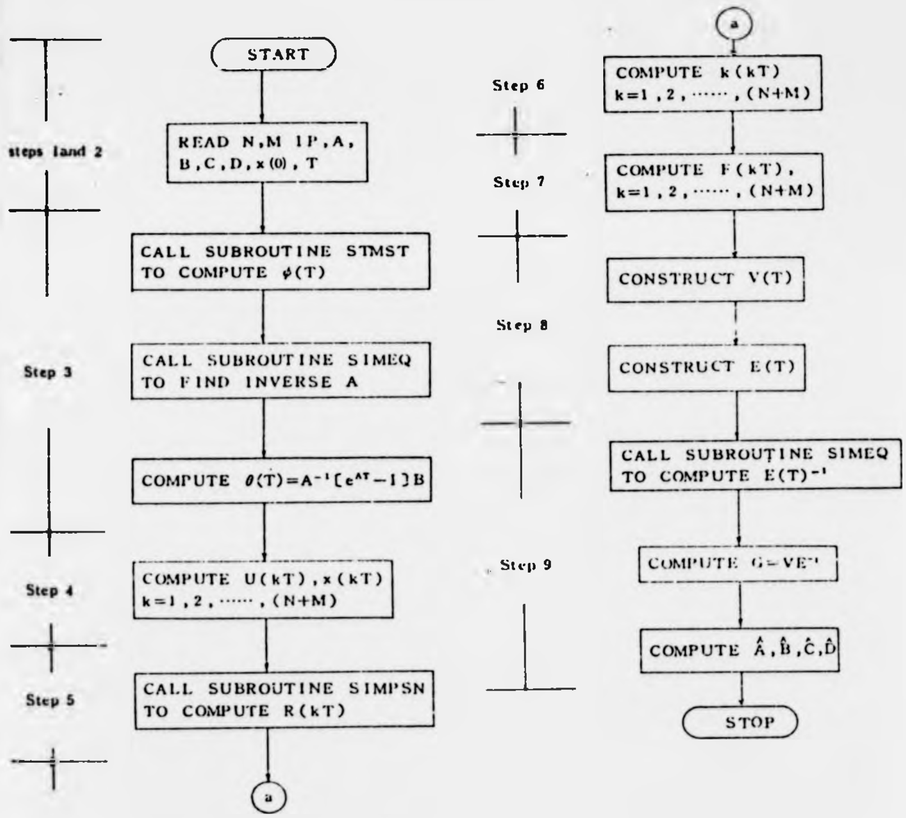

Fig. 2—1 The flow-chart of computer program to identify multiple-input multiple-output known order linear time-invariant systems

(a) <u>Example 2 — 1</u> The system considered here to demonstrate the validity of the above algorithm has the following transfer function.

$$\begin{bmatrix} y_1(s) \\ \\ y_2(s) \end{bmatrix} = \underbrace{\begin{bmatrix} \dfrac{2s^2+7s+8}{s^3+6s^2+11s+6} & \dfrac{s^2+2s}{s^3+6s^2+11s+6} \\ \dfrac{s^2+4s+3}{s^3+6s^2+11s+6} & \dfrac{2s^2+8s+6}{s^3+6s^2+11s+6} \end{bmatrix}}_{\text{" true } G(s) \text{"}} \begin{bmatrix} u_1(s) \\ \\ u_2(s) \end{bmatrix}$$

In this example G(s) is treated as the physical system of known order but of unknown parameters. To distinguish the above "physical system" (simulated for the illustration here) from the "identified system", the above system is referred to here as the " true system " as opposed to the (approximate) identified system. The step responses of the true system are shown in Fig. 2 — 2 and its state-space representation is

$$\underbrace{\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix}}_{\dot{\underline{x}}(t)} = \underbrace{\begin{bmatrix} -1 & 1 & 0 \\ 0 & -4 & 1 \\ 0 & -2 & -1 \end{bmatrix}}_{\text{" true A "}} \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}}_{\underline{x}(t)} + \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 0 & -1 \end{bmatrix}}_{\text{" true B "}} \underbrace{\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}}_{\underline{u}(t)}$$

$$\underbrace{\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}}_{\underline{y}(t)} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & -1 \end{bmatrix}}_{\text{"true C"}} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\text{"true D"}} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

The general identification configuration is shown in Fig. 2-3, where sinusoidal input signals are used to generate the output responses needed for the identification algorithm. For this illustrative example, the physical system as well as the identification hardware were simulated on the digital computer. The data from the "hardware" were used as the input measured data in to the identification software. For an arbitrary choice of T = 10 mseconds, the identified parameters generated by the software are

$$\hat{A} = \begin{bmatrix} -1.0000+D00 & 1.0000\ D+00 & -2.1980\ D-07 \\ -7.1530\ D-07 & -4.0000\ D+00 & 1.0000\ D+00 \\ -1.6090\ D-06 & -2.0000\ D+00 & -1.0000\ D+00 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} 1.0000\ D+00 & -1.3040\ D-07 \\ 1.0000\ D+00 & 1.0000\ D+00 \\ 6.5570\ D-07 & -1.0000\ D+00 \end{bmatrix}$$

$$\hat{C} = \begin{bmatrix} 1.0000\ D+00 & 1.0000\ D+00 & 3.5760\ D-07 \\ 3.0990\ D-06 & 1.0000\ D+00 & -1.0000\ D+00 \end{bmatrix}$$

$$\hat{D} = \begin{bmatrix} 5.9600\,D-08 & 3.5760\,D-07 \\ -1.3710\,D-06 & 7.7490\,D-07 \end{bmatrix}$$

The step responses for the identified system are shown in Fig. 2-4 which demonstrate the general validity of the numerical algorithm developed here.

Although fairly good agreement between the true parameters and the identified parameters is obtained in this algorithm, the extent of error introduced in the identification procedure is very much dependent on the choice of T, the sampling interval. In general, the Nyquist sampling theorem* could be used here to choose T. However, in a completely automated identification scheme, the natural frequency or the waveforms of the measurement (continuous) signals may not be known. Although several established identification algorithms use very small sampling intervals, there is no established guidelines on the choice of T (apart from the hardware/measurement constraints imposed by a particular interface circuitry). Since the main characteristics of the proposed algorithm are its simplicity and compatibility with low-cost computer systems, the sampling time is likely to be much larger than what is available on mainframe interface hardware. For this reason, there is

---

* this is normally used in the context of data capturing rather than parameter identification (further comments on P2-29.)

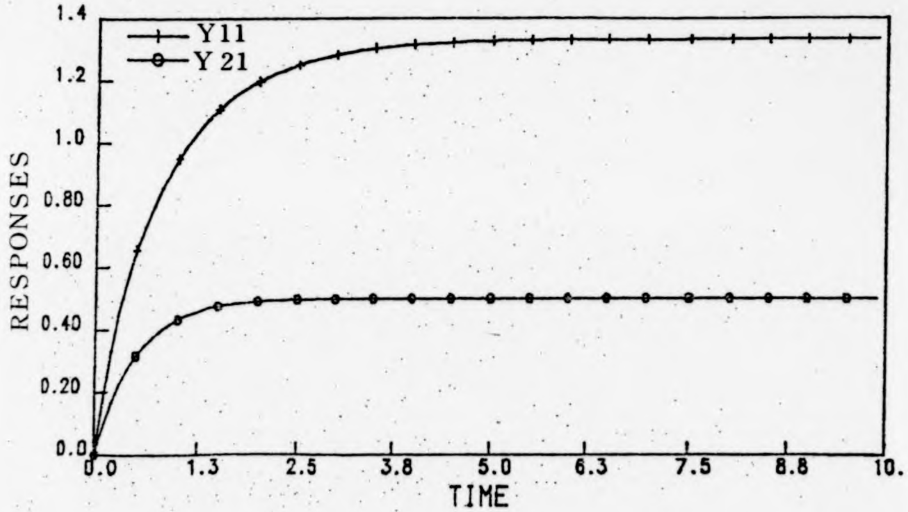

Fig. 2 − 2 The step responses of the true system for $u_1(t) = 1.0$, $u_2(t) = 0.0$

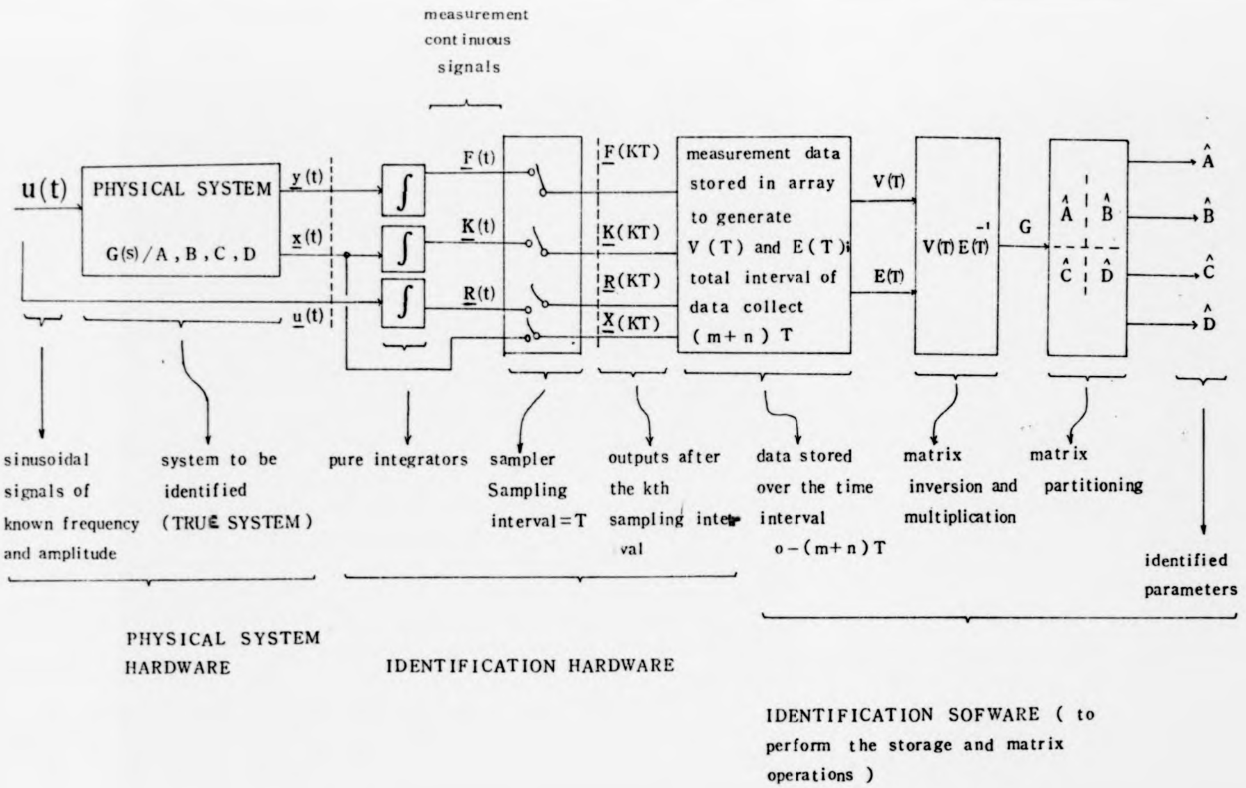

Fig. 2-3 The general identification configuration

a need to study the effect of sampling interval on the estimation error.

To provide a quantitative measure of estimation error the following scalar error indices are defined :

$$S_A = \sum_{\substack{i=1\\j=1}}^{n} (a_{ij} - \hat{a}_{ij})^2 / n \times n$$

$$S_B = \sum_{\substack{i=1\\k=1}}^{n,m} (b_{ik} - \hat{b}_{ik})^2 / n \times m$$

$$S_C = \sum_{\substack{\ell=1\\i=1}}^{p,n} (c_{\ell i} - \hat{c}_{\ell i})^2 / n \times p$$

$$S_D = \sum_{\substack{\ell=1\\k=1}}^{p,m} (d_{\ell k} - \hat{d}_{\ell k})^2 / m \times p$$

$$S = \frac{S_A + S_B + S_C + S_D}{4}$$

$$\left.\begin{array}{l} A = \{ a_{ij} \} \\ \hat{A} = \{ \hat{a}_{ij} \} \\ B = \{ b_{ik} \} \\ \hat{B} = \{ \hat{b}_{ik} \} \\ C = \{ c_{li} \} \\ \hat{C} = \{ \hat{c}_{li} \} \\ D = \{ d_{lk} \} \\ \hat{D} = \{ \hat{d}_{lk} \} \\ i \in 1, \cdots, n \\ j \in 1, \cdots, n \\ k \in 1, \cdots, m \\ \ell \in 1, \cdots, p \end{array}\right. \qquad (2-48)$$

Where A , B , C , D represented the parameters of the true system and $\hat{A}$, $\hat{B}$, $\hat{C}$, $\hat{D}$ the corresponding estimated parameters.

Since the dominant time-costants are commonly used in assessing the transient behaviour of linear system, it is useful to relate the choice of the sampling interval to the dominant time constant (T_C) or the undamped oscillation time period

(T_o), that is

$$\left.\begin{aligned} T &= K\,T_c \quad \text{for a damped system} \\ &= K\,T_o \quad \text{for an undamped system} \end{aligned}\right\} \qquad (2-49)$$

where K is a constant.

To assess the effect of the choice of T on the estimation error, the following third order systems with two-inputs two-outputs and similar structure are considered here:

(a) overdamped

(b) underdamped

(c) undamped

(d) unstable

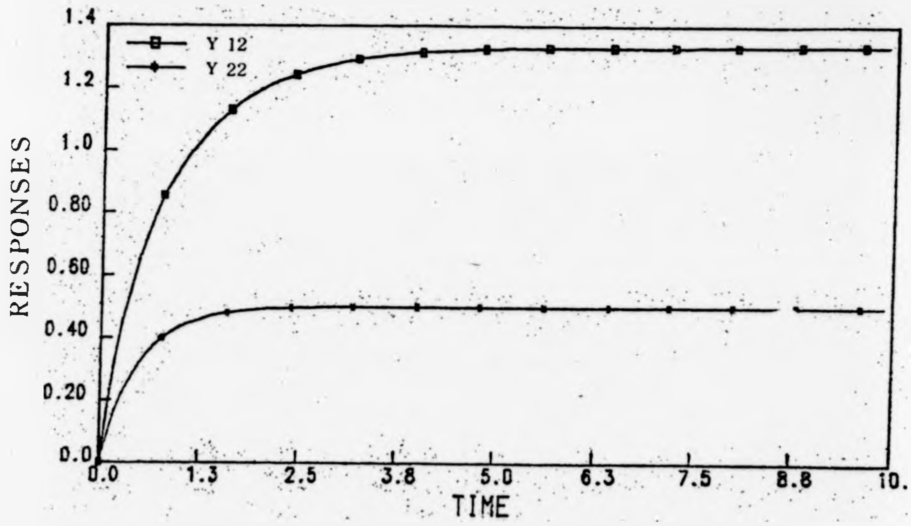

Fig. 2－4 The step responses of the identified system for $u_1(t) = 1.0$, $u_2(t) = 0.0$

(b) <u>Example 2-2 overdamped system</u>: The true system is represented by

$$G(s)=\begin{bmatrix} \dfrac{5.75s+23.575}{s^3+4.1s^2+3.4s+0.3} & \dfrac{4}{s^3+4.1s^2+3.4s+0.3} \\ \dfrac{1.15s^2+4.715s}{s^3+4.1s^2+3.4s+0.3} & \dfrac{0.8s}{s^3+4.1s^2+3.4s+0.3} \end{bmatrix}$$

for which

$$A=\begin{bmatrix} 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 1.00 \\ -0.30 & -3.40 & -4.10 \end{bmatrix}, \quad B=\begin{bmatrix} 0.00 & 0.00 \\ 2.30 & 0.00 \\ 0.00 & 1.60 \end{bmatrix}$$

$$C=\begin{bmatrix} 2.50 & 0.00 & 0.00 \\ 0.00 & 0.50 & 0.00 \end{bmatrix}, \quad D=\begin{bmatrix} 0.00 & 0.00 \\ 0.00 & 0.00 \end{bmatrix}$$

The step responses of this true system (overdamped) are shown in Fig. 2-5 which indicate that the system has a dominant time-constant of around 10 seconds. The system is controllable and the two input-output channels are linearly independent, and hence E(T) in equation (2-34) will be non-singular. Consequently, the identification algorithm developed earlier is applicable.

Applying the procedure indicated by Fig. 2-3 for a known order system, the choice of sampling interval T=1 second ($T_c/10$) gives the following estimation for the system

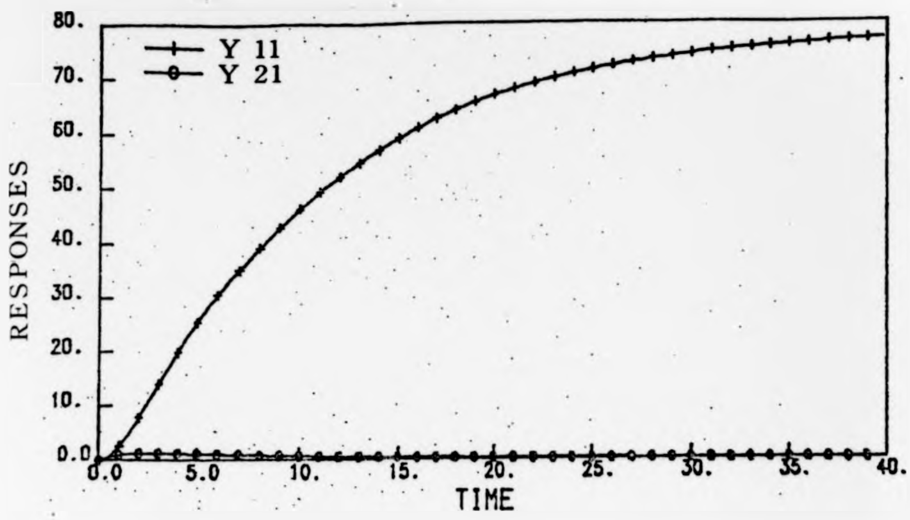

Fig. 2－5　The step responses of the true system (overdamped) for $u_1(t) = 1.0$, $u_2(t) = 0.0$

parameters:

$$\hat{A} = \begin{bmatrix} 0.2220\,D-15 & 0.1000\,D+01 & -0.3553\,D-14 \\ -0.2290\,D-14 & -0.2753\,D-13 & 0.1000\,D+01 \\ -0.3000\,D+00 & -0.3400\,D+01 & -0.4100\,D+01 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} -0.3553\,D-14 & 0.2887\,D-14 \\ 0.2300\,D+01 & 0.1754\,D-13 \\ 0.3553\,D-14 & 0.1600\,D+01 \end{bmatrix}$$

$$\hat{C} = \begin{bmatrix} 0.2500\,D+01 & -0.2842\,D-12 & 0.2274\,D-12 \\ 0.2776\,D-16 & 0.5000\,D+00 & -02220\,D-14 \end{bmatrix}$$

$$\hat{D} = \begin{bmatrix} -0.1137\,D-12 & -0.2274\,D-12 \\ -0.8882\,D-15 & 0.2220\,D-15 \end{bmatrix}$$

The estimation errors (2 — 48) for this choice of T are:

$$S_A = 0.8620\,D-28$$

$$S_B = 0.5687\,D-28$$

$$S_C = 0.2208\,D-25$$

$$S_D = 0.1616\,D-25$$

$$S = 0.9596\,D-26$$

which are acceptable for most analysis/design studies using linear theory ; the agreement between the step responses (Fig. 2 — 7) of the true system (Fig. 2 — 5) and estimated system (Fig. 2 — 6) is also within acceptable error limits.

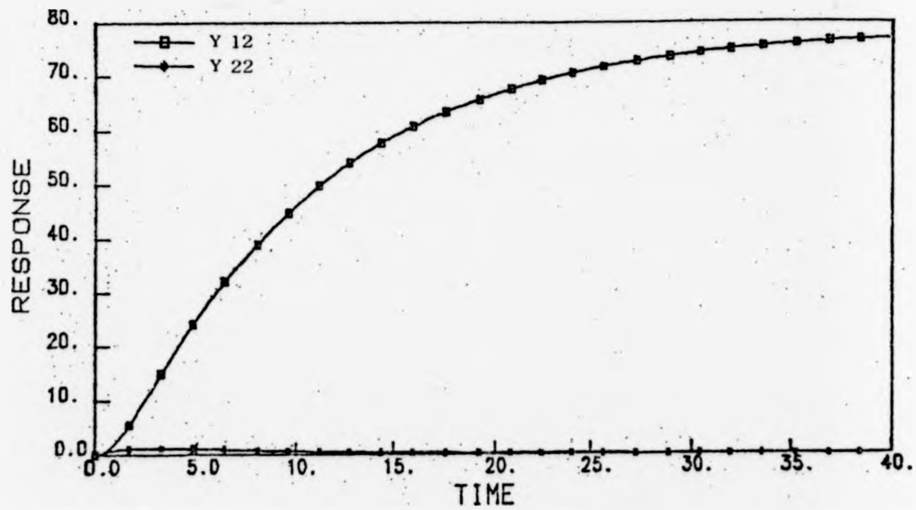

Fig. 2 — 6 The step responses of the identified system (overdamped) for $u_1(t) = 1.0$, $u_2(t) = 0.0$

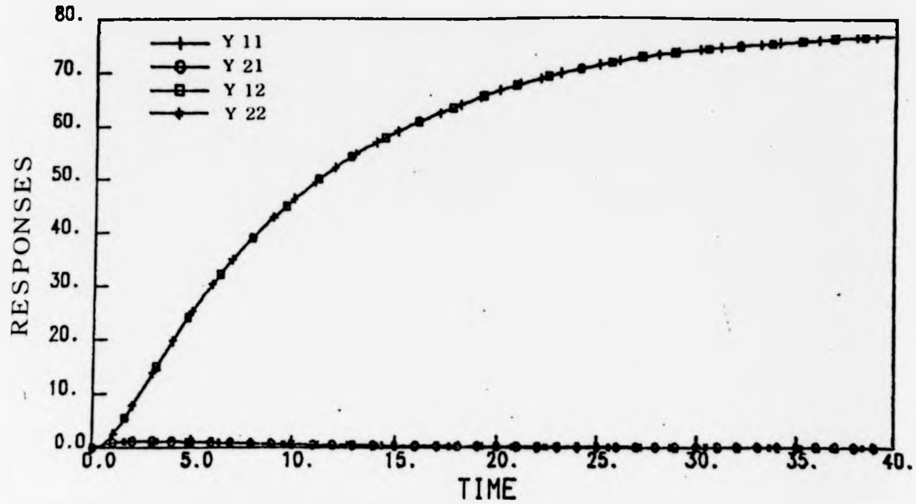

Fig. 2—7 The superimposed step responses of the true and the identified system (overdamped) for $u_1(t) = 1.0$, $u_2(t) = 0.0$

To study the effect of the choice of the sampling interval (T) on the estimation errors (S_A , S_B , S_C , S_D , S) , T was varied from 1 msecond to 6 seconds 〔 K in (2—49) varying from 0.0001 to 0.6 〕, the resulting errors are ploted in Fig. 2—8 . The following observation can be made from these error curves :

(1) Square of the estimation errors are small in amplitude and within the acceptable bound for most design procedures over a very wide sampling intervals (T = 1 msecond to 5.0 seconds). For the system in Example 2—2 , the "optimum" choice is around 1 second which is one-tenth of the dominant time constant of the system (T_C = 10 seconds).

(2) Since there is no high-frequency leading or trailing edges in the system response, the sampling interval for "data capturing" according to Nyquist sampling theorem should be around 5 seconds. This is five fold more than the sampling interval needed for parameter identification. Although this single example is not sufficient to derive any general conclusions, it is worth noting that the sampling interval predicted by the Nyquist sampling theorem may not be most appropriate for parameter identification algorithms.

(3) It is interesting to observe that the estimation errors are higher at the two boundaries. The estimation algo-

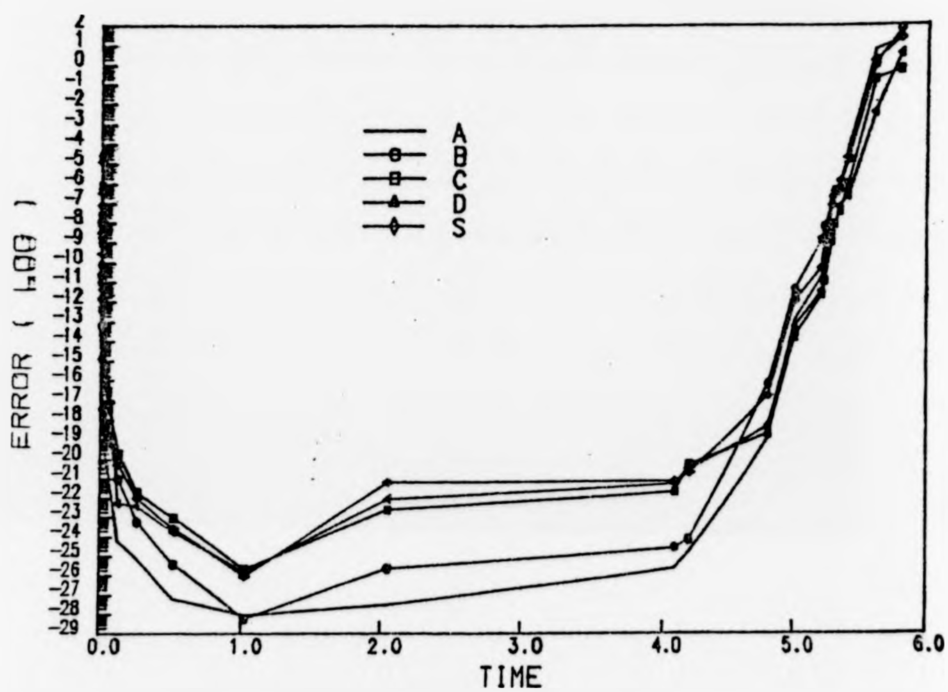

Fig. 2 — 8 The effect of varying T on the estimation errors (overdamped)

rithm is based the mathematical theorem which constructs the two matrice V (T) (2 − 45) and E (T) (2 − 46) out of the (m + n) T samples of each parameter $\underline{x}(t)$, $\int \underline{x}(t)$, $\underline{y}(t)$, $\int \underline{y}(t)$, $\underline{u}(t)$, and $\int \underline{u}(t)$, the mathematics of the algorithm is based an processing these 6 sets of (m+n) T discrete data and consequently the total time interval over which measurement data is collected is fixed when T is assigned a particular value.

Thus for smaller values of T, measurement data are collected over a smaller period of time (with m and n having specific values). For the example considered above, a choice of $T = 1$ msecond allows only (5×1) mseconds for data collection, since this is well below the settling time of the system, a considerable amount of error is identified parameters is to be expected.

At the other end of the spectrum, a large choice of T, e.g., $T = 5$ seconds, gives a period of (5×5) seconds which is well above the settling time of the system. However, the samples are taken at 5 seconds interval which allows five samples over the settling time of the system. This, for most engineering systems, is not adequate to capture the true transient response, and consequently the error introduced in identification is high.

The choice of $T = 1$ second takes 5 samples during the settling period of the system. For the overdamped system this is adequate and this is reflected in the smaller band of errors in Fig. 2 — 8. Although the sampling interval necessary for acceptable identification errors has been evaluated in this example through numerical results, for physical systems an a-priori knowledge of its time-response/time-constant is unlikely to be available. Since the total number of samples to be taken is dictated by the mathematics, the sampling interval T may be chosen to fit in most of the measurement period over the settling time of the system.

For the third-order two-input two-output system, the number of measurement data is $(3+2)\ T = 5\ T$. For a choice of $T = 1$ second there is a need to store only 5×5 measurement data. Although this measurement data storage requirement is very modest, there is a need for a very large amount of space for matrix manipulation as indicated by Fig. 2 — 3. For the example considered here, a total of 21 Kbytes of memory (IBM — 4331, 32-bit word length data) was required for data storage and numerical computation. The total CPU time needed to compute the identification for this example was around 1.4 seconds. For small computers e.g. IBM-PC (IBM — 5550 model), the storage requirement is not a problem, but the estimation error may increase due to shorter word length (typically, 16-bit) unless the program was modified to perform calculations with 2-byte words. The processing time needed to compute the algorithm will increase with smaller computers due to architechtural constraints on data and address buses.

Having established the general validity of the new identification algorithm, the following three examples are considered to find out if there is any relationship between the choice of sampling interval and the transient behaviour of the true system, the three systems considered are underdamped, undamped and unstable—all having similar structure as the previous example.

(c) <u>Example 2 — 3 underdamped system :</u> The true system is represented by

$$G(s)=\begin{bmatrix} \dfrac{3.45s+41.4}{s^3+12s^2+25s+50} & \dfrac{2.4}{s^3+12s^2+25s+50} \\ \dfrac{12.42s^2+149.04s}{s^3+12s^2+25s+50} & \dfrac{8.64s}{s^3+12s^2+25s+50} \end{bmatrix}$$

The step responses are shown in Fig. 2 — 9 , which suggest that the dominant time constant of the system is around 1 second. and the settling time is around 5 seconds. For this system , the total number of measurement data needed for each variable is (m + n) T = 5 T. Allowing for 5 samples during the settling time period of the system, this gives an identification sampling interval (T) of 0.1 second (Tc/10).

The identified parameters given by the algorithm for T = 0.1 seconds are :

$$\hat{A} = \begin{bmatrix} -0.5107\,D-14 & 0.1000\,D+01 & -0.1110\,D-14 \\ -0.4547\,D-12 & -0.5258\,D-12 & 0.1000\,D+01 \\ -0.5000\,D+02 & -0.2500\,D+02 & -0.1200\,D+02 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} -0.6661\,D-15 & 0.1466\,D-15 \\ 0.2300\,D+01 & 0.9076\,D-14 \\ 0.2842\,D-12 & 0.1600\,D+01 \end{bmatrix}$$

$$\hat{C} = \begin{bmatrix} 0.1500\,D+01 & -0.3979\,D-12 & -0.6750\,D-13 \\ -0.3908\,D-13 & 0.5400\,D+01 & -0.2220\,D-14 \end{bmatrix}$$

$$\hat{D} = \begin{bmatrix} 0.1421\,D-13 & -0.2220\,D-15 \\ -0.1110\,D-14 & 0.7772\,D-15 \end{bmatrix}$$

The step responses of this identified system is shown in Fig. 2-10. For the true system, the parameters are :

$$A = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ -50.0 & -25.0 & -12.0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0 & 0.0 \\ 2.3 & 0.0 \\ 0.0 & 1.6 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.5 & 0.0 & 0.0 \\ 0.0 & 5.4 & 0.0 \end{bmatrix}, \quad D = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}$$

Thus the identification errors (2-49) are

$$S_A = 0.5369\,D-21$$

$$S_B = 0.1348\,D-22$$

$$S_C = 0.2740\,D-22$$

$$S_D = 0.5095\,D-23$$

$$S = 0.2366\,D-22$$

The effects of varying T on S_A, S_B, S_C, S_D and S are shown in Fig. 2-11, which supports the comments made at the end of Example 2-2. The agreement between the step responses of the true system and the identified system (Fig. 2-12) is also within acceptable error limits.

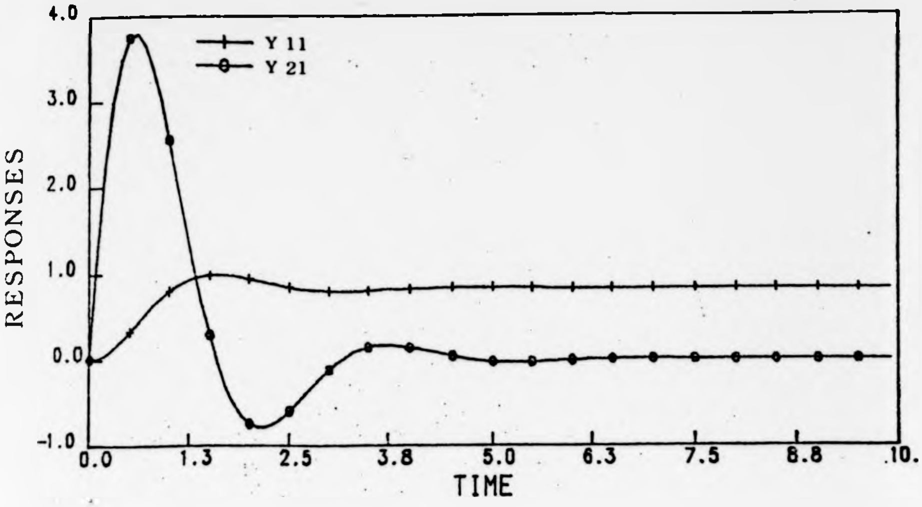

Fig. 2 − 9 The step responses of the true system (underdamped) for $u_1(t)=1.0$, $u_2(t)=0.0$

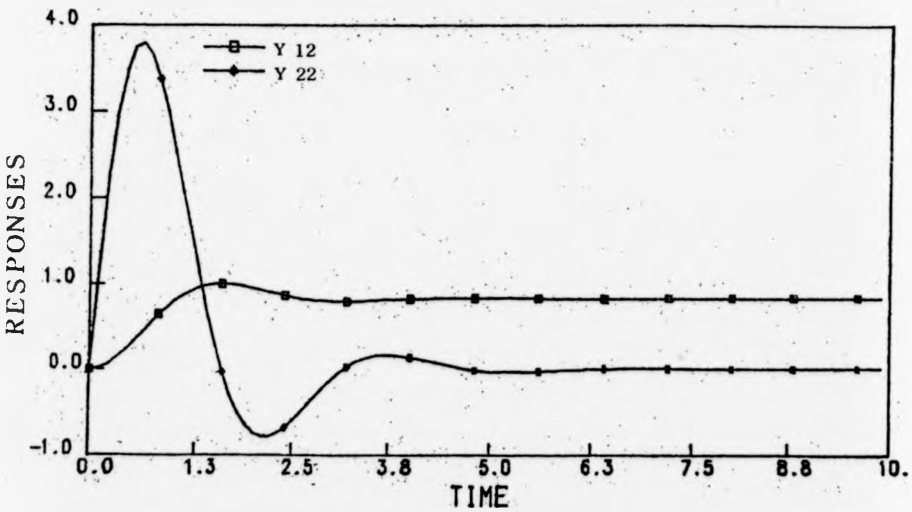

Fig. 2−10 The step responses of the identified system (underdamped) for $u_1(t)=1.0$, $u_2(t)=0.0$

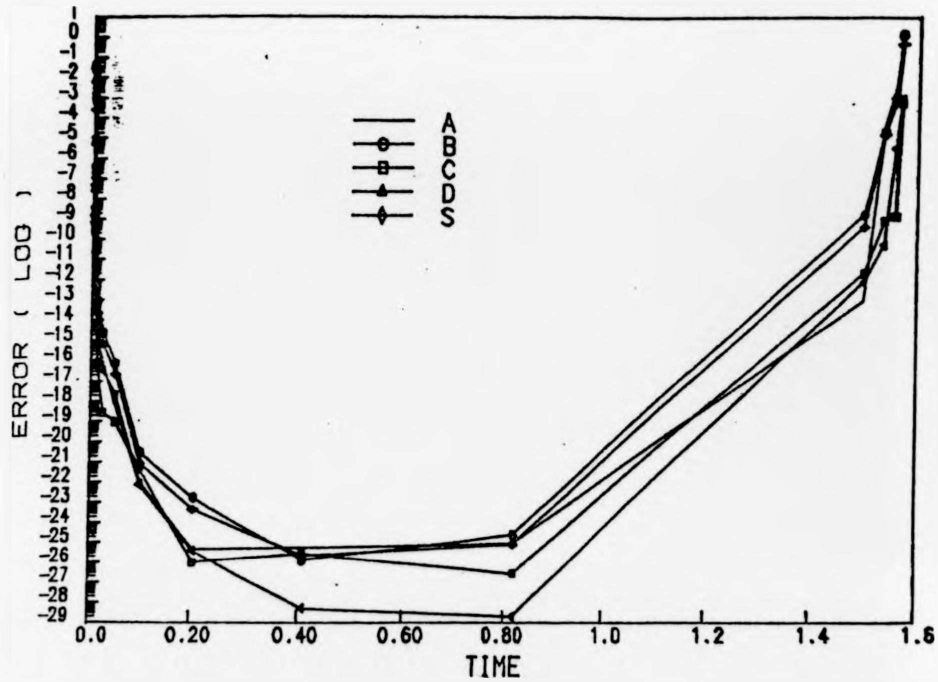

Fig. 2-11 The effects of varying T on identification errors (underdamped)

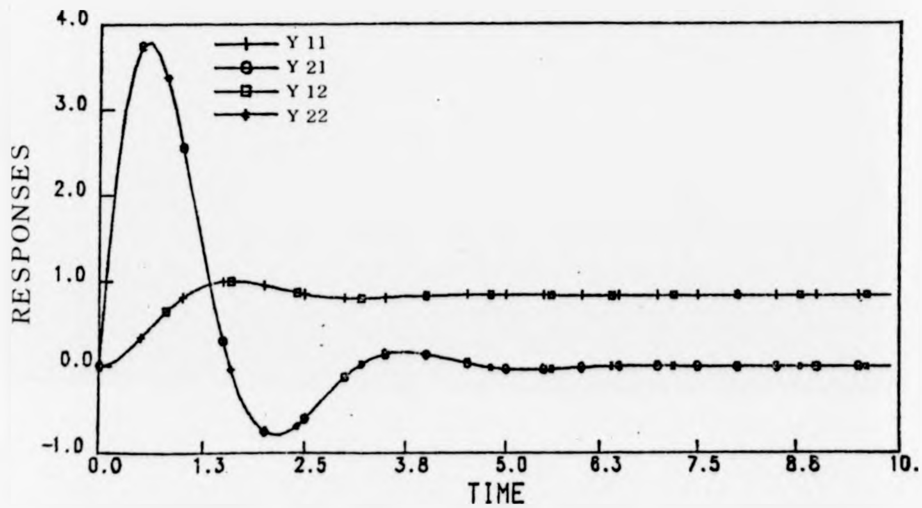

Fig. 2-12 The superimposed step responses of the true and the identified system (underdamped) for $u_1(t) = 1.0$, $u_2(t) = 0.0$

(d) Example 2-4 undamped system : The true system is represented by

$$G(s) = \begin{bmatrix} \dfrac{2s+40}{s^3+20s^2+100s+2000} & \dfrac{1}{s^3+20s^2+100s+2000} \\ \dfrac{-9s^2+180s}{s^3+20s^2+100s+2000} & \dfrac{4.5s}{s^3+20s^2+100s+2000} \end{bmatrix}$$

The step responses are shown in Fig. 2-13 , which suggest that the oscillation time period $T_0 = 0.628$ seconds. For this system the total number of measurement data needed for each variable is $(m+n)T = 5T$, we choose the identification sampling interval (T) as 0.0628 seconds ($T_0/10$).

The identified parameters given by the algorithm for $T = 0.0628$ seconds are :

$$\hat{A} = \begin{bmatrix} 0.0000\,D+00 & 1.0000\,D+00 & 3.5530\,D-14 \\ 3.6380\,D-12 & 2.5930\,D-13 & 1.0000\,D+00 \\ -2.0000\,D+03 & -1.0000\,D+02 & -2.0000\,D+01 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} -1.0660\,D-14 & 5.7730\,D-15 \\ 2.0000\,D+00 & 9.5920\,D-14 \\ -1.0910\,D-11 & 1.0000\,D+00 \end{bmatrix}$$

$$\hat{C} = \begin{bmatrix} 1.0000\,D+00 & 3.5530\,D-15 & 3.5530\,D-15 \\ 2.1830\,D-11 & 4.5000\,D+00 & 1.7050\,D-13 \end{bmatrix}$$

$$\hat{D} = \begin{bmatrix} -3.5530\,D-15 & 8.8820\,D-16 \\ -7.8160\,D-14 & 2.1320\,D-14 \end{bmatrix}$$

The step responses of this identified system are shown in Fig. 2—14.

For the true system, the parameters are:

$$A = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ -2000.0 & -100.0 & -20.0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0 & 0.0 \\ 2.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 4.5 & 0.0 \end{bmatrix}, \quad D = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}$$

Thus the identification errors (2—49) are:

$$S_A = 0.1478\,D-23$$

$$S_B = 0.1884\,D-22$$

$$S_C = 0.7943\,D-22$$

$$S_D = 0.1644\,D-26$$

$$S = 0.2519\,D-22$$

The effects of varying T on S_A, S_B, S_C, S_D and S are shown in Fig. 2—15, which supports the comments made at the end of Example 2—2. The agreement between the step responses of the true system and the identified system (Fig. 2—16) is also within acceptable error limits.

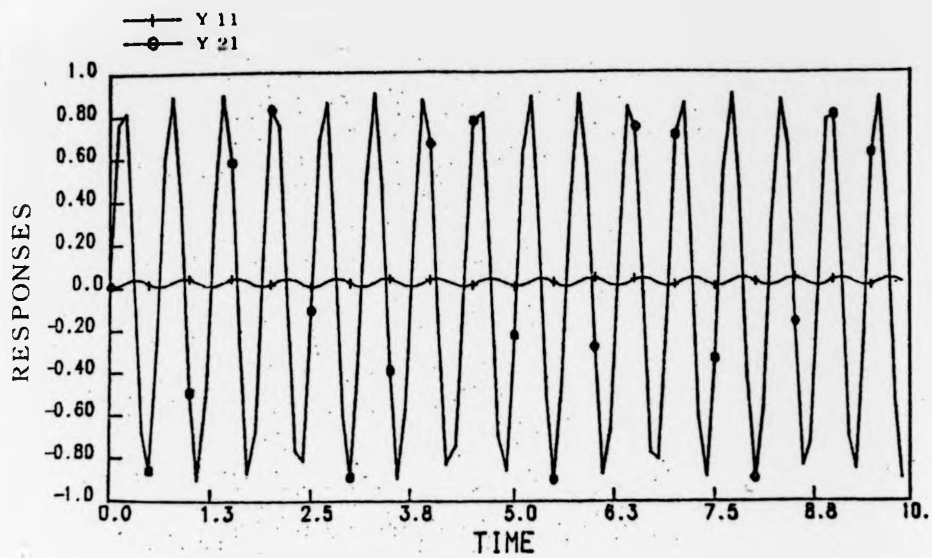

Fig. 2−13 The step responses of the true system (undamped) for $u_1(t)=1.0$, $u_2(t)=0.0$

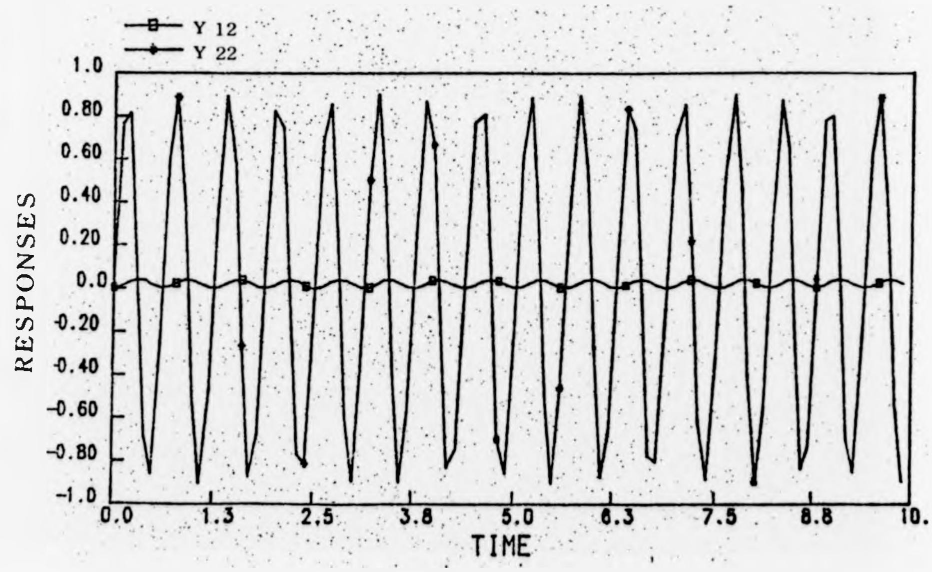

Fig. 2−14 The step responses of the identified system (undamped) for $u_1(t)=1.0$, $u_2(t)=0.0$

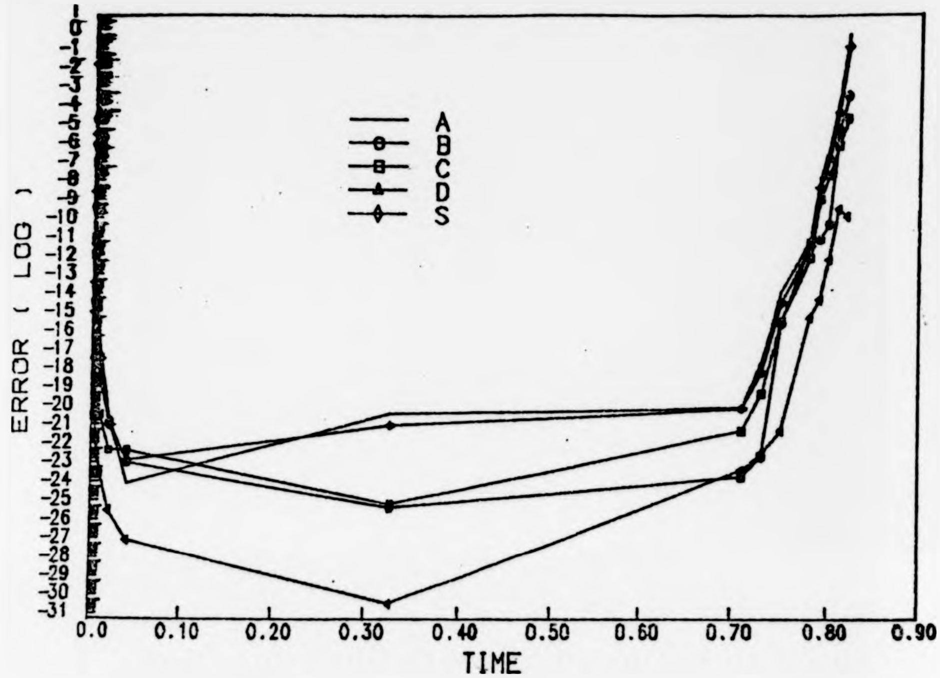

Fig. 2 — 15 The effects of varying T on identification errors (undamped)

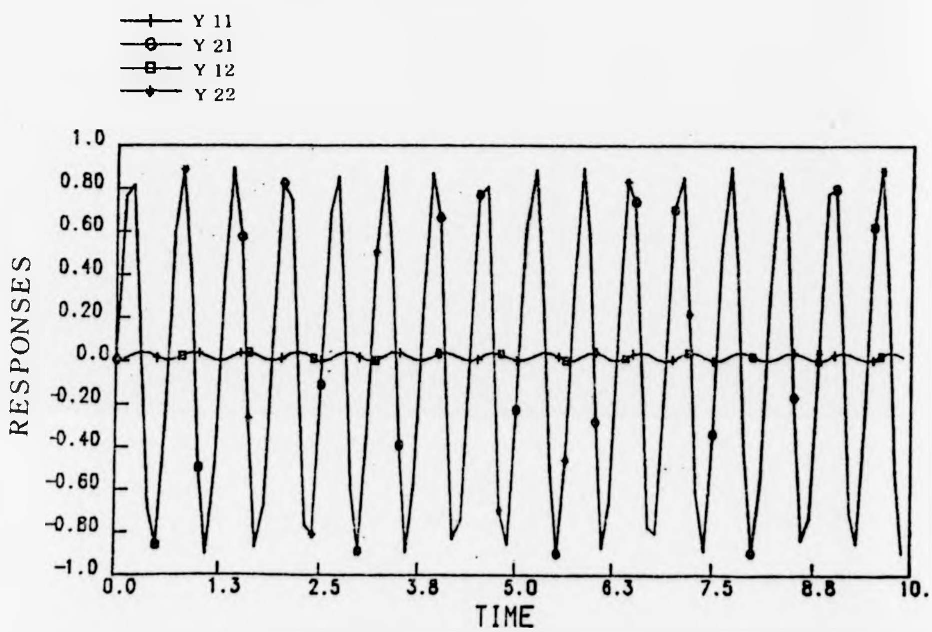

Fig. 2 — 16 The superimposed step responses of the true and identified system (undamped) for $u_1(t) = 1.0$, $u_2(t) = 0.0$

(e) Example 2 − 5 unstable system : The true system is represented by

$$G(s) = \begin{bmatrix} \dfrac{2s+69}{s^3+34.5s^2+282.5s-150} & \dfrac{1}{s^3+34.5s^2+282.5s-150} \\ \dfrac{5s^2+172.5s}{s^3+34.5s^2+282.5s-150} & \dfrac{2.5s}{s^3+34.5s^2+282.5s-150} \end{bmatrix}$$

The step responses are shown in Fig. 2 − 17, these suggest that the dominant time constant of the system is around 2 seconds. For this system the total number of measurement data needed for each variable is $(m+n)T = 5T$, we choose the identification sampling interval (T) is 0.2 seconds ($T_c/10$).

The identified parameters given by the algorithm for T = 0.2 seconds are

$$\hat{A} = \begin{bmatrix} 1.8190\,D-12 & 1.0000\,D+00 & -6.2530\,D-13 \\ 1.4550\,D-11 & 7.2760\,D-11 & 1.0000\,D+00 \\ 1.5000\,D+02 & -2.8250\,D+02 & -3.4500\,D+01 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} 1.0230\,D-12 & -2.8420\,D-14 \\ 2.0000\,D+00 & 4.5470\,D-13 \\ 6.9850\,D-10 & 1.0000\,D+00 \end{bmatrix}$$

$$\hat{C} = \begin{bmatrix} 1.0000\,D+00 & 1.7050\,D-13 & 3.5530\,D-15 \\ 9.0950\,D-13 & 2.5000\,D+00 & -1.4780\,D-12 \end{bmatrix}$$

$$\hat{D} = \begin{bmatrix} 3.1970\,D-14 & -2.2200\,D-16 \\ 9.0950\,D-13 & -7.8169\,D-14 \end{bmatrix}$$

The step responses of this identified system are shown in Fig. 2—18.

For the true system, the parameters are :

$$A = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ 150.0 & -282.5 & -34.5 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0 & 0.0 \\ 2.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 2.5 & 0.0 \end{bmatrix}, \quad D = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}$$

Thus the identification errors (2—48) are

$$S_A = 0.6122\,D-20$$

$$S_B = 0.8132\,D-19$$

$$S_C = 0.8068\,D-22$$

$$S_D = 0.2086\,D-24$$

$$S = 0.2048\,D-19$$

The effects of varying T on S_A, S_B, S_C, S_D and S are shown in Fig. 2—19, which supports the comments made at the end of Example 2—2. The agreement between the step responses of the true system and the identified system (Fig. 2—20) is also within acceptable error limits.

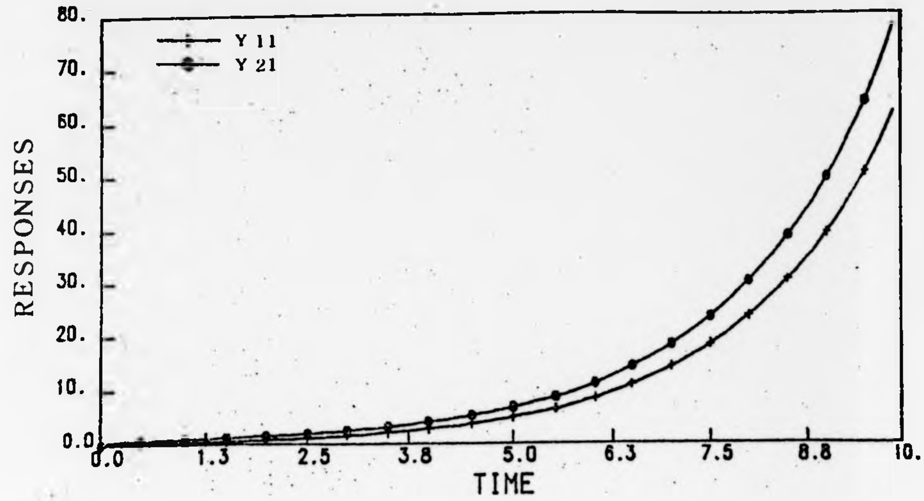

Fig. 2 — 17 The responses of the true system (unstable) for $u_1(t)=1.0$, $u_2(t)=0.0$

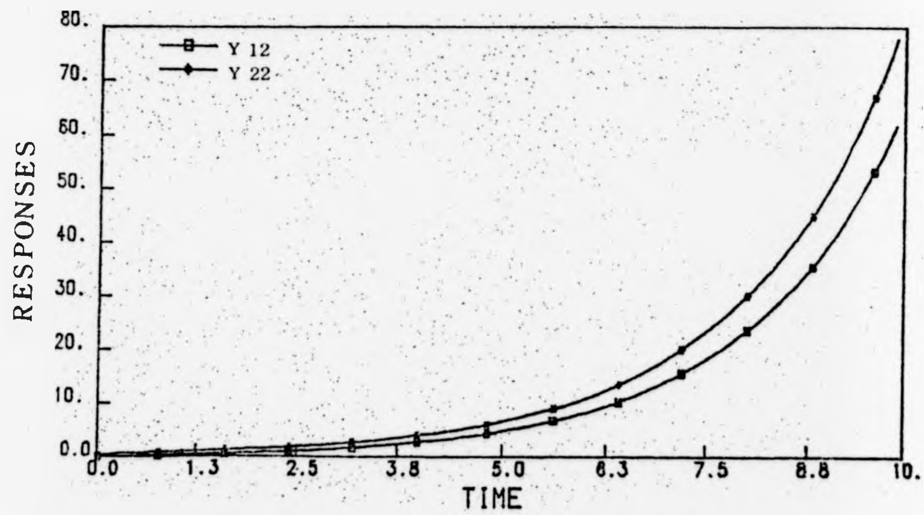

Fig. 2 — 18 The step responses of the identified system (unstable) for $u_1(t)=1.0$, $u_2(t)=0.0$

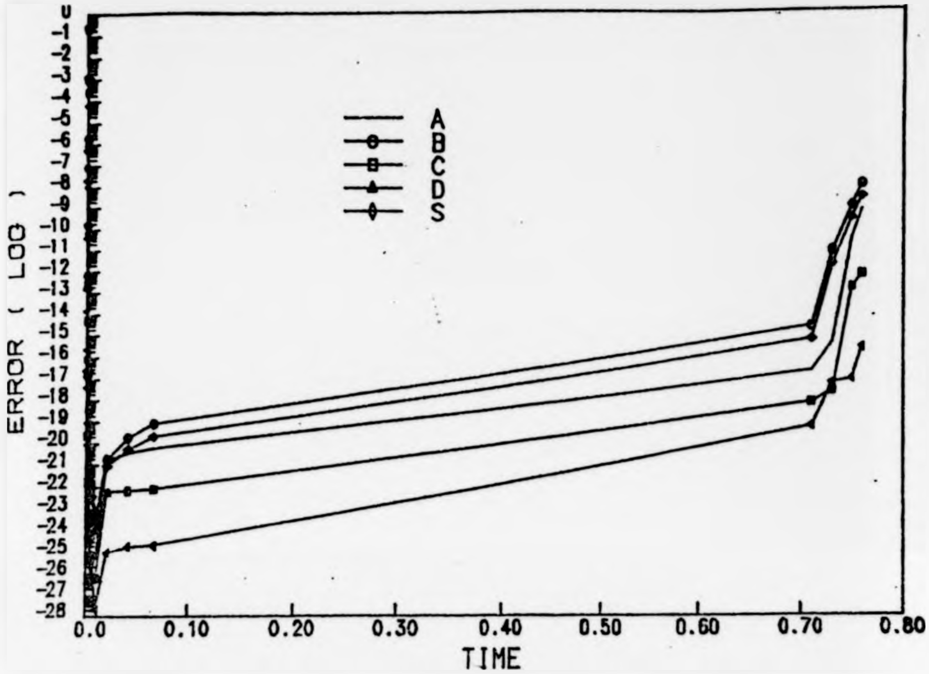

Fig.2 — 19 The effects of varying T on identification errors (unstable)

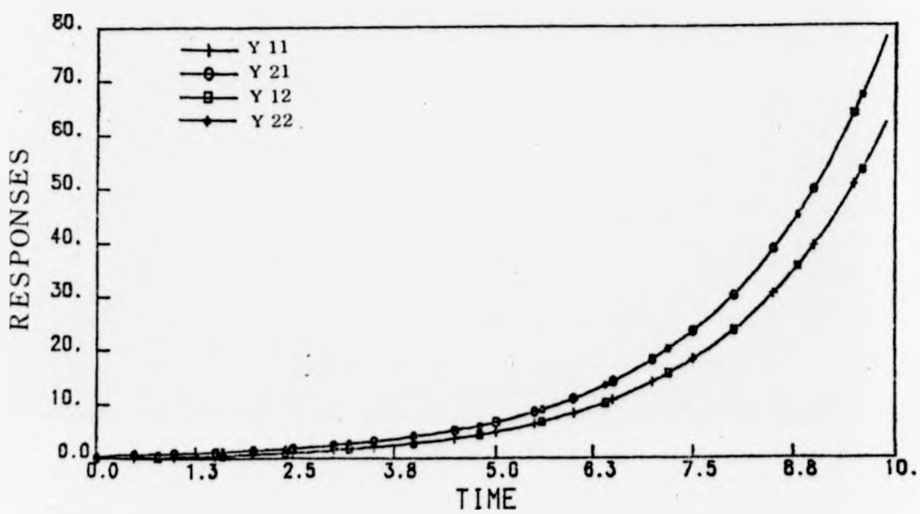

Fig. 2 — 20 The superimposed step responses of the true and the identified system (unstable)for $u_1(t)= 1.0$, $u_2(t)=0.0$

2–3 Concluding remarks

The mathematical results and the associated numerical algorithm developed in this chapter have been found to be satisfactory for the identification of systems with a wide range of transient characteristics. The numerical procedure has been observed to be able to cope with systems of orders up to 10 without any problems of convergence. This method is based on the integration of state and output equations; this reduces the effects of system noise. In this method no special test signal is required, which makes the method applicable for on-line identification as well as estimation of parameter variations which may occur during the normal operation of the system under consideration. Despite this and the fact that the choice of the sampling interval is not difficult, this method is not generally acceptable. This is due to the fact that this method is based on an a-priori knowledge of the system order. The problem of identification with unknown system order is considered in the following chapter.

Chapter — 3 UNKNOWN ORDER MIMO SYSTEMS

The preceding chapter established a relatively simple method of estimating the parameters of a multiple-input multiple-output (MIMO) time-invariant continuous system of known order. These results are extended in this chapter to the cases where the system order is not known. The method developed here is based on a special structure of the system matrix A and multiple-integration of the dynamical equations. The main advantage of the method developed here is that it requires output data and not state variables, as was the case in Chapter 2.

The system $\bar{S}$ considered here is described by

$$\bar{S}: \quad \dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t) \tag{3—1}$$

$$\underline{y}(t) = C\,\underline{x}(t) \tag{3—2}$$

where $\underline{x}(t) \in R^{n \times 1}$ is the state vector

$\underline{x}(t_0) \in R^{n \times 1}$ is the initial state vector

$\underline{y}(t) \in R^{p \times 1}$ is the output vector

$\underline{u}(t) \in R^{m\times 1}$ is the input vector

$A \in R^{n\times n}$, $B \in R^{n\times m}$ and $C \in R^{p\times n}$ are real constant matrices

Let

$D[n_a(m+2)] \triangleq \{ \underline{u}(k) \in R^{m\times 1}, \underline{y}(k) \in R^{p\times 1} ; k = 1, 2, \cdots n_a(m+2)\}$

be a sequence of $n_a(m+2)$ input-output measurements from the system $\overline{S}$. where n_a is the assumed system order (n_a is an initial, usually lower, value chosen by the algorithm).

The task is to use the above finite input-output measurements $D[n_a(m+2)]$, to derive an estimate for the system order and parameters.

Consider the system $\overline{S}$ given by (3−1) and (3−2), where we assume that only the input $\underline{u}(t)$ and the output $\underline{y}(t)$ are accessible for measurement.

The identification method of the continuous systems is based on the following special system representation :

(i) A is of the form:

$$A = \begin{array}{c} \begin{array}{cc} \overbrace{\hphantom{0 \; 0 \cdots\cdots\cdots 0}}^{(p)\text{ columns}} & \overbrace{\hphantom{1 \; 0 \; 0 \cdots\cdots 0}}^{(n-p)\text{ columns}} \end{array} \\ \left[\begin{array}{cccc|cccc} 0 & 0 & \cdots & 0 & 1 & 0 & 0 \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & 0 \cdots & 0 \\ \vdots & & & \vdots & \vdots & & & \\ 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots 0 & 1 \\ \hline a_{(n-p+1)1} & & a_{(n-p+1)2} & \cdots & \cdots & & & a_{(n-p+1)n} \\ \vdots & & \vdots & & & & & \vdots \\ a_{n1} & & a_{n2} & \cdots & \cdots & & & a_{nn} \end{array} \right] \end{array} \begin{array}{l} \left.\vphantom{\begin{array}{c}0\\0\\ \vdots\\0\end{array}}\right\} (n-p)\text{ rows} \\ \left.\vphantom{\begin{array}{c}a\\ \vdots\\a\end{array}}\right\} (p)\text{ rows} \end{array}$$

$$= \left[\begin{array}{c|c} O_{(n-p)\times p} & I_{(n-p)\times(n-p)} \\ \hline \multicolumn{2}{c}{\bar{A}_{p\times n}} \end{array} \right] \qquad (3-3)$$

(ii) C is of the form:

$$C = \begin{array}{c} \overbrace{\hspace{6em}}^{(p)\text{ columns}} \quad \overbrace{\hspace{6em}}^{(n-p)\text{ columns}} \\ \left[\begin{array}{cccc|ccc} 1 & 0 & 0 \cdots\cdots & 0 & 0 & 0 \cdots\cdots & 0 \\ 0 & 1 & 0 \cdots\cdots & 0 & 0 & 0 \cdots\cdots & 0 \\ 0 & 0 & 1 \cdots\cdots & 0 & \vdots & & \vdots \\ \vdots & & & & & & \\ 0 & 0 & 0 \cdots\cdots & 1 & 0 & \cdots\cdots & 0 \end{array}\right] \end{array}$$

$$= \left[\, I_{p\times p} \quad O_{p\times(n-p)} \,\right] \qquad (3-4)$$

(iii) The form of B is arbitrary.

The given system $\overline{S}$ can be transformed into the special representation given by (3–3) and (3–4) as can be seen in the following Lemma :

Lemma 3–1 Given the system $\overline{S}$, where A is $n\times n$ matrix, C is $p\times n$ matrix. Let $\overline{S}$ to be completely observable and rank(C) = p.

define :

$$E = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_p \\ \vdots \\ E_{2p} \\ \vdots \\ E_n \end{bmatrix} \qquad (3-5)$$

where

$$\begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_p \end{bmatrix} = C, \quad \begin{bmatrix} E_{p+1} \\ E_{p+2} \\ \vdots \\ E_{2p} \end{bmatrix} = CA, \cdots\cdots, \quad \begin{bmatrix} E_{ip+1} \\ E_{ip+2} \\ \vdots \\ E_{(i+1)p} \end{bmatrix} = CA^i \qquad (3\text{-}6)$$

where

$E \in R^{n\times n}$, $E_i^T \in R^{n\times 1}$

$ip+j=n$, $j=1,2,\cdots,p$ and $i=0,1,2,\cdots,\rho$ where ρ is the least integer such that $\rho \leq n/p$.

Define a new state $\underline{z}(t)=E\underline{x}(t)$ and transform the system $\overline{S}$ into the canonical form, since $\overline{S}$ is completely observable and rank$(C)=p$, hence E is non-singular, then:

$$\begin{aligned} \dot{\underline{z}}(t) &= EAE^{-1}\underline{z}(t)+EB\underline{u}(t) \\ &= \hat{A}\underline{z}(t)+\hat{B}\underline{u}(t) \end{aligned} \qquad (3\text{-}7)$$

$$\begin{aligned} \underline{y}(t) &= CE^{-1}\underline{z}(t) \\ &= \hat{C}\underline{z}(t) \end{aligned} \qquad (3\text{-}8)$$

where

$$\hat{A} = EAE^{-1} = \left[\begin{array}{c|c} O_{(n-p)\times p} & I_{(n-p)\times(n-p)} \\ \hline \multicolumn{2}{c}{\overline{A}_{p\times n}} \end{array}\right] \qquad (3\text{-}9)$$

$$\hat{C} = CE^{-1} = [\, I_{p\times p} \mid O_{p\times(n-p)} \,] \qquad (3\text{-}10)$$

$$\hat{B} = EB \qquad (3\text{-}11)$$

$$\hat{\underline{x}}(t_0) = E\underline{x}(t_0) \qquad (3\text{-}12)$$

proof:

$$\hat{C}E = [\, I_{p\times p} \mid O_{p\times(n-p)} \,] \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_p \\ \vdots \\ E_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_p \end{bmatrix} = C$$

$$EA = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix} \cdot A = \begin{bmatrix} E_1 A \\ E_2 A \\ \vdots \\ E_n A \end{bmatrix}$$

$$\hat{A}E = \left[\begin{array}{ccc} O_{(n-p)\times p} & \multicolumn{2}{|c}{I_{(n-p)\times(n-p)}} \\ \hline a_{(n-p+1)1} & a_{(n-p+1)2} \cdots\cdots & a_{(n-p+1)n} \\ a_{(n-p+2)1} & a_{(n-p+2)2} \cdots\cdots & a_{(n-p+2)n} \\ \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} \cdots\cdots & a_{nn} \end{array}\right] \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_p \\ \vdots \\ E_{n-1} \\ E_n \end{bmatrix}$$

$$= \begin{bmatrix} E_{p+1} \\ E_{p+2} \\ \vdots \\ E_{2p} \\ \vdots \\ E_{n-p} \\ a_{(n-p+1)1}E_1 + a_{(n-p+1)2}E_2 + \cdots\cdots + a_{(n-p+1)n}E_n \\ \vdots \\ a_{n1}E_1 + a_{n2}E_2 + \cdots\cdots + a_{nn}E_n \end{bmatrix}$$

Since E is non-singular, the row vectors $E_1, E_2, \cdots\cdots, E_n$ are linearly independent. So we can form the linear combinations of these independent vectors, if we pick a_{ij} ; $i=(n-p+1), \cdots, n$ and $j=1,2,\cdots,n$ not all zero such that

$$\sum_{j=1}^{n} a_{(n-p+1)j} E_j = E_{n-p+1}$$

$$\sum_{j=1}^{n} a_{(n-p+2)j} E_j = E_{n-p+2}$$

$$\vdots$$

$$\sum_{j=1}^{n} a_{nj} E_j = E_n$$

then

$$\hat{A}E = \begin{bmatrix} E_{p+1} \\ E_{p+2} \\ \vdots \\ E_{2p} \\ \vdots \\ E_{n-p} \\ E_{n-p+1} \\ \vdots \\ E_n \end{bmatrix} = EA$$

hence

$$\hat{A} = EAE^{-1}$$

<u>Proposition 3-1</u> : Let $A \in R^{n\times n}$, $C \in R^{p\times n}$ and

$$Q = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{3-13}$$

where $Q \in R^{np \times n}$

$$C = [\, I_{p\times p} \mid O_{p\times(n-p)} \,] \tag{(3-4)}$$

$$A = \left[\begin{array}{c|c} O_{(n-p)\times p} & I_{(n-p)\times(n-p)} \\ \hline \multicolumn{2}{c}{\bar{A}_{p\times n}} \end{array}\right] \tag{(3-3)}$$

Now if the system matrix A and the output matrix C of the system $\bar{S}$ are in the form of (3-3)、(3-4) respectively, then Q is of the form :

$$Q = \left[\begin{array}{c} I_{n\times n} \\ \hline O_{(np-n)\times n} \end{array}\right] \tag{3-14}$$

Furthermore if the constant matrix S is of the form:

$$S \triangleq [\, I_{n\times n} \mid O_{n\times(np-n)} \,] \tag{3-15}$$

where

$S \in R^{n\times np}$

then

$$SQ = I_{n\times n} \tag{3-16}$$

3 — 1 Identification from input-output data

In the case where the order of the system to be identified is not known a-priori then the following results provide a method to determine the system order n.

Let the actual system order be n and the assumed system order be n_a and consider the system $\overline{S}$ represented by equations (3-1) and (3-2).

Define :

$$\begin{bmatrix} x_{11}(t) \\ x_{12}(t) \\ \vdots \\ x_{1n}(t) \end{bmatrix} \triangleq \int_{t_0}^{t} \begin{bmatrix} x_1(\tau) \\ x_2(\tau) \\ \vdots \\ x_n(\tau) \end{bmatrix} d\tau$$

or

$$\underline{x}_1(t) \triangleq \int_{t_0}^{t} \underline{x}(\tau)\, d\tau \tag{3-17}$$

$$\begin{bmatrix} x_{(i+1)1}(t) \\ x_{(i+1)2}(t) \\ \vdots \\ x_{(i+1)n}(t) \end{bmatrix} \triangleq \int_{t_0}^{t} \begin{bmatrix} x_{i1}(\tau) \\ x_{i2}(\tau) \\ \vdots \\ x_{in}(\tau) \end{bmatrix} d\tau$$

or

$$\underline{x}_{i+1}(t) \triangleq \int_{t_0}^{t} \underline{x}_i(\tau)\, d\tau \tag{3-18}$$

$$\begin{bmatrix} u_{11}(t) \\ u_{12}(t) \\ \vdots \\ u_{1m}(t) \end{bmatrix} \triangleq \int_{t_0}^{t} \begin{bmatrix} u_1(\tau) \\ u_2(\tau) \\ \vdots \\ u_m(t) \end{bmatrix} d\tau$$

or

$$\underline{u}_1(t) = \int_{t_0}^{t} \underline{u}(\tau)\, d\tau \tag{3-19}$$

$$\begin{bmatrix} u_{(i+1)1}(t) \\ u_{(i+1)2}(t) \\ \vdots \\ u_{(i+1)m}(t) \end{bmatrix} \triangleq \int_{t_0}^{t} \begin{bmatrix} u_{i1}(\tau) \\ u_{i2}(\tau) \\ \vdots \\ u_{im}(\tau) \end{bmatrix} d\tau$$

or

$$\underline{u}_{i+1}(t) \triangleq \int_{t_0}^{t} \underline{u}_i(\tau)\, d\tau \qquad (3\text{—}20)$$

$$\begin{bmatrix} y_{11}(t) \\ y_{12}(t) \\ \vdots \\ y_{1p}(t) \end{bmatrix} \triangleq \int_{t_0}^{t} \begin{bmatrix} y_1(\tau) \\ y_2(\tau) \\ \vdots \\ y_p(\tau) \end{bmatrix} d\tau$$

or

$$\underline{y}_1(t) \triangleq \int_{t_0}^{t} \underline{y}(\tau)\, d\tau \qquad (3\text{—}21)$$

$$\begin{bmatrix} y_{(i+1)1}(t) \\ y_{(i+1)2}(t) \\ \vdots \\ y_{(i+1)p}(t) \end{bmatrix} \triangleq \int_{t_0}^{t} \begin{bmatrix} y_{i1}(\tau) \\ y_{i2}(\tau) \\ \vdots \\ y_{ip}(\tau) \end{bmatrix} d\tau$$

or

$$\underline{y}_{i+1}(t) \triangleq \int_{t_0}^{t} \underline{y}_i(\tau)\, d\tau \qquad (3\text{—}22)$$

Now if we integrate both sides of equation (3—1) n_a times, n_a-1 time……, we get :

$\underline{n_a \text{ times}}$:

$$\underline{x}_{n_a-1}(t) - \frac{(t-t_0)^{n_a-1}}{(n_a-1)!}\underline{x}(t_0) = A\underline{x}_{n_a}(t) + B\underline{u}_{n_a}(t) \qquad (3\text{—}23)$$

$\underline{n_a - 1 \text{ times}}$:

$$\underline{x}_{n_a-2}(t) - \frac{(t-t_0)^{n_a-2}}{(n_a-2)!}\underline{x}(t_0) = A\underline{x}_{n_a-1}(t) + B\underline{u}_{n_a-1}(t) \qquad (3\text{—}24)$$

$\vdots$

$\underline{2 \text{ integrations}}$:

$$\underline{x}_1(t) - (t-t_0)\underline{x}(t_0) = A\underline{x}_2(t) + B\underline{u}_2(t) \qquad (3\text{—}25)$$

Integrating both sides of (3—2) n_a times, n_a-1 times ······ and using equation (3—23), we have :

$\underline{n_a \text{ times}}$:

$$\underline{y}_{n_a}(t) = C\underline{x}_{n_a}(t)$$

$n_a - 1$ times

$$\underline{y}_{n_a-1}(t) = C\underline{x}_{n_a-1}(t)$$

$$= CA\underline{x}_{n_a}(t) + CB\underline{u}_{n_a}(t) + \frac{(t-t_0)^{n_a-1}}{(n_a-1)!}C\underline{x}(t_0)$$

$\underline{n_a - 2 \text{ times}}$:

$$\underline{y}_{n_a-2}(t) = C\underline{x}_{n_a-2}(t)$$

$$= CA^2\underline{x}_{n_a}(t) + CAB\underline{u}_{n_a}(t) + CB\underline{u}_{n_a-1}(t)$$

$$+ \frac{(t-t_0)^{n_a-1}}{(n_a-1)!}CA\underline{x}(t_0) + \frac{(t-t_0)^{n_a-2}}{(n_a-2)!}C\underline{x}(t_0)$$

$\vdots$

(3—26)

1 integration :

$$\underline{y}_1(t) = C\underline{x}_1(t)$$

$$= CA^{n_a-1}\underline{x}_{n_a}(t) + CA^{n_a-2}B\underline{u}_{n_a}(t) + \cdots\cdots + CB\underline{u}_2(t)$$

$$+ \frac{(t-t_0)^{n_a-1}}{(n_a-1)!}CA^{n_a-2}\underline{x}(t_0) + \cdots\cdots + (t-t_0)C\underline{x}(t_0)$$

Rearranging in the matrix form, we have

$$\begin{bmatrix} \underline{y}_{n_a}(t) \\ \underline{y}_{n_a-1}(t) \\ \vdots \\ \underline{y}_1(t) \end{bmatrix} = \begin{bmatrix} C\underline{x}_{n_a}(t) \\ CA\underline{x}_{n_a}(t) \\ \vdots \\ CA^{n_a-1}\underline{x}_{n_a}(t) \end{bmatrix} + \begin{bmatrix} O_{p\times 1} \\ CB\underline{u}_{n_a}(t) \\ \vdots \\ CA^{n_a-2}B\underline{u}_{n_a}(t) \end{bmatrix} + \cdots + \begin{bmatrix} O_{(n_a-1)p\times 1} \\ \hdashline CB\underline{u}_2(t) \end{bmatrix}$$

$$+ \begin{bmatrix} O_{p\times 1} \\ C\underline{x}(t_0) \\ CA\underline{x}(t_0) \\ \vdots \\ CA^{n_a-2}\underline{x}(t_0) \end{bmatrix} \cdot \frac{(t-t_0)^{n_a-1}}{(n_a-1)!} + \begin{bmatrix} O_{2p\times 1} \\ C\underline{x}(t_0) \\ CA\underline{x}(t_0) \\ \vdots \\ CA^{n_a-3}\underline{x}(t_0) \end{bmatrix} \frac{(t-t_0)^{n_a-2}}{(n_a-2)!} + \cdots\cdots$$

$$+ \cdots + \begin{bmatrix} O_{(n_a-1)p\times 1} \\ \hdashline C\underline{x}(t_0) \end{bmatrix} (t-t_0) \qquad (3\text{—}27)$$

Define :

$$\underline{E}(t) \triangleq \begin{bmatrix} \underline{y}_{n_a}(t) \\ \underline{y}_{n_a-1}(t) \\ \vdots \\ \underline{y}_1(t) \end{bmatrix} \qquad (3\text{—}28)$$

$$\underline{F}(t)=\begin{bmatrix} \underline{u}_{n_a}(t) \\ \underline{u}_{n_a-1}(t) \\ \vdots \\ \underline{u}_1(t) \end{bmatrix} \tag{3-29}$$

and

$$\underline{G}(t)=\begin{bmatrix} \dfrac{(t-t_0)^{n_a-1}}{(n_a-1)!} \\ \dfrac{(t-t_0)^{n_a-2}}{(n_a-2)!} \\ \vdots \\ (t-t_0) \\ 1 \end{bmatrix} \tag{3-30}$$

where

$\underline{E}(t)\in R^{n_a p\times 1}$

$\underline{F}(t)\in R^{n_a m\times 1}$

$\underline{G}(t)\in R^{n_a\times 1}$

Define :

$$Q_0 \triangleq Q=\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n_a-1} \end{bmatrix}, \quad Q_1=\begin{bmatrix} O_{p\times n_a} \\ C \\ CA \\ \vdots \\ CA^{n_a-2} \end{bmatrix}, \cdots\cdots$$

$$Q_i=\begin{bmatrix} O_{ip\times n_a} \\ \vdots \\ C \\ CA \\ \vdots \\ CA^{n_a-1-i} \end{bmatrix} \tag{3-31}$$

$$Q_{n_a} = [\,0\,] \tag{3-32}$$

Define

$$J \triangleq [\, Q_1 B \;\; Q_2 B \cdots\cdots Q_{n_a} B \,] \tag{3-33}$$

$$H \triangleq [\, Q_1 \underline{x}(t_0) \;\; Q_2 \underline{x}(t_0) \cdots\cdots Q_{n_a} \underline{x}(t_0) \,] \tag{3-34}$$

where

$$J \in R^{n_a p \times n_a m}, \; H \in R^{n_a p \times n_a}$$

Using the above definitions, equation (3—27) can be put in a matrix form as

$$\underline{E}(t) = Q\underline{x}_{n_a}(t) + J\underline{F}(t) + H\underline{G}(t) \tag{3-35}$$

Multiplying both sides of (3—35) by the matrix S, we get :

$$S\underline{E}(t) = SQ\underline{x}_{n_a}(t) + SJ\underline{F}(t) + SH\underline{G}(t) \tag{3-36}$$

From proposition 3—1 , we have :

$$SQ = I_{n_a \times n_a}$$

then

$$S\underline{E}(t) = \underline{x}_{n_a}(t) + SJ\underline{F}(t) + SH\underline{G}(t) \tag{3-37}$$

If we differentiate both sides of equation (3—37) we get:

$$S\dot{\underline{E}}(t) = \underline{x}_{n_a-1}(t) + SJ\dot{\underline{F}}(t) + SH\dot{\underline{G}}(t) \tag{3-38}$$

where

$$\dot{\underline{E}}(t) = \begin{bmatrix} \underline{y}_{n_a-1}(t) \\ \underline{y}_{n_a-2}(t) \\ \vdots \\ \underline{y}_1(t) \\ \underline{y}(t) \end{bmatrix}$$

$$\dot{\underline{x}}_{n_a}(t) = \underline{x}_{n_a-1}(t)$$

$$\underline{\dot{F}}(t)=\begin{bmatrix} \underline{u}_{n_a-1}(t) \\ \underline{u}_{n_a-2}(t) \\ \vdots \\ \underline{u}_1(t) \\ \underline{u}(t) \end{bmatrix}$$

and

$$\underline{\dot{G}}(t)=\begin{bmatrix} \frac{(t-t_0)^{n_a-2}}{(n_a-2)!} \\ \vdots \\ (t-t_0) \\ 1 \\ 0 \end{bmatrix}$$

then

$$S\underline{\dot{E}}(t)=A\underline{x}_{n_a}(t)+B\underline{u}_{n_a}(t)+\frac{(t-t_0)^{n_a-1}}{(n_a-1)!}\underline{x}(t_0)$$

$$+SJ\underline{\dot{F}}(t)+SH\underline{\dot{G}}(t) \tag{3-39}$$

substituting for $\underline{x}_{n_a}(t)$ from (3−37), we get:

$$S\underline{\dot{E}}(t)=AS\underline{E}(t)-ASJ\underline{F}(t)-ASH\underline{G}(t)+B\underline{u}_{n_a}(t)+SJ\underline{\dot{F}}(t)$$

$$+SH\underline{\dot{G}}(t)+\frac{(t-t_0)^{n_a-1}}{(n_a-1)!}\underline{x}(t_0) \tag{3-40}$$

we have

$$SJ\underline{\dot{F}}(t)+B\underline{u}_{n_a}(t)=SJ\underline{\dot{F}}(t)+SQB\underline{u}_{n_a}(t)$$

$$=S[J\underline{\dot{F}}(t)+QB\underline{u}_{n_a}(t)]$$

we have

$$J\underline{\dot{F}}(t)+QB\underline{u}_{n_a}(t)=$$

$$\begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ CB & 0 & & 0 & 0 \\ CAB & CB & & \vdots & \vdots \\ \vdots & \vdots & & 0 & 0 \\ CA^{n_a-2}B & CA^{n_a-3}B & \cdots & CB & 0 \end{bmatrix} \begin{bmatrix} \underline{u}_{n_a-1}(t) \\ \underline{u}_{n_a-2}(t) \\ \vdots \\ \underline{u}_1(t) \\ \underline{u}(t) \end{bmatrix} + \begin{bmatrix} CB \\ CAB \\ \vdots \\ CA^{n_a-2}B \\ CA^{n_a-1}B \end{bmatrix} \underline{u}_{n_a}(t)$$

$$= \begin{bmatrix} CB\underline{u}_{n_a}(t) \\ CAB\underline{u}_{n_a}(t)+CB\underline{u}_{n_a-1}(t) \\ CA^2B\underline{u}_{n_a}(t)+CAB\underline{u}_{n_a-1}(t)+CB\underline{u}_{n_a-2}(t) \\ \vdots \\ CA^{n_a-1}B\underline{u}_{n_a}(t)+CA^{n_a-2}B\underline{u}_{n_a-1}(t)+CA^{n_a-3}B\underline{u}_{n_a-2}(t)+\cdots+CB\underline{u}_1(t) \end{bmatrix}$$

$$=K\underline{F}(t) \qquad (3\text{—}41)$$

where

$$K \triangleq [\, Q_0B \;\; Q_1B \cdots\cdots Q_{n_a-1}B \,]$$

$$K \in R^{n_a p \times n_a m}$$

then

$$SJ\underline{\dot{F}}(t)+B\underline{u}_{n_a}(t)=SK\underline{F}(t) \qquad (3\text{—}42)$$

Also, we have

$$SH\underline{\dot{G}}(t)+\frac{(t-t_0)^{n_a-1}}{(n_a-1)!}\underline{x}(t_0)$$

$$=S[\,H\underline{\dot{G}}(t)+\frac{(t-t_0)^{n_a-1}}{(n_a-1)!}Q\underline{x}(t_0)\,]$$

we have

$$H\dot{\underline{G}}(t)+Q\underline{x}(t_0)\frac{(t-t_0)^{n_a-1}}{(n_a-1)!}$$

$$=\begin{bmatrix} 0 & 0 \cdots\cdots & 0 & 0 \\ C\underline{x}(t_0) & 0 & 0 & 0 \\ CA\underline{x}(t_0) & C\underline{x}(t_0) & \vdots & \vdots \\ \vdots & \vdots & 0 & 0 \\ CA^{n_a-2}\underline{x}(t_0) & CA^{n_a-3}\underline{x}(t_0)\cdots\cdots & C\underline{x}(t_0) & 0 \end{bmatrix}\cdot$$

$$\begin{bmatrix} \dfrac{(t-t_0)^{n_a-2}}{(n_a-2)!} \\ \dfrac{(t-t_0)^{n_a-3}}{(n_a-3)!} \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} C\underline{x}(t_0) \\ CA\underline{x}(t_0) \\ CA^2\underline{x}(t_0) \\ \vdots \\ CA^{n_a-1}\underline{x}(t_0) \end{bmatrix}\frac{(t-t_0)^{n_a-1}}{(n_a-1)!}$$

$$=[\,Q_0\underline{x}(t_0)\quad Q_1\underline{x}(t_0)\cdots\cdots Q_{n_a-1}\underline{x}(t_0)]\begin{bmatrix} \dfrac{(t-t_0)^{n_a-1}}{(n_a-1)!} \\ \dfrac{(t-t_0)^{n_a-2}}{(n_a-2)!} \\ \vdots \\ 1 \end{bmatrix}$$

$$=W\underline{G}(t) \tag{3-43}$$

where

$$W\triangleq[\,Q_0\underline{x}(t_0)\quad Q_1\underline{x}(t_0)\cdots\cdots Q_{n_a-1}\underline{x}(t_0)] \tag{3-44}$$

$$W\in R^{n_a p\times n_a}$$

Thus

$$S H \underline{\dot{G}}(t)+\frac{(t-t_0)^{n_a-1}}{(n_a-1)!}\underline{x}(t_0)=SW\underline{G}(t) \tag{3-45}$$

So now equation (3-40) becomes

$$S\underline{\dot{E}}(t)=AS\underline{E}(t)-ASJ\underline{F}(t)+SK\underline{F}(t)-ASH\underline{G}(t)+SW\underline{G}(t)$$

Define :

$$P\triangleq SK-ASJ \tag{3-46}$$

$$M\triangleq SW-ASH \tag{3-47}$$

$$P\in R^{n_a\times n_a m}\ ,\quad M\in R^{n_a\times n_a}$$

Then

$$S\underline{\dot{E}}(t)=AS\underline{E}(t)+P\underline{F}(t)+M\underline{G}(t) \tag{3-48}$$

Let

$$\underline{R}(t)=S\underline{E}(t)$$

Where

$$\underline{R}(t)\in R^{n_a\times 1}$$

In a partitioned form equation (3-48) can be written as :

$$\underline{\dot{R}}(t)=[\,A\ \vdots\ P\ \vdots\ M\,]\begin{bmatrix}\underline{R}(t)\\ \underline{F}(t)\\ \underline{G}(t)\end{bmatrix} \tag{3-49}$$

Define :

$$\underline{V}(t)\triangleq\begin{bmatrix}\underline{R}(t)\\ \underline{F}(t)\\ \underline{G}(t)\end{bmatrix} \tag{3-50}$$

where

$$\underline{V}(t) \in R^{n_a(m+2)\times 1}$$

then

$$\dot{\underline{R}}(t) = [\, A \mid P \mid M \,]\, \underline{V}(t) \tag{3-51}$$

Now if $\underline{y}(t)$, $\underline{u}(t)$, $\underline{y}_i(t)$ and $\underline{u}_i(t)$ $i = 1, 2, \cdots, n_a$ are measured at $n_a(m+2)$ successive samples of time with a sampling period T, then from (3-51) we have :

$$\dot{R}(T) = [\, A \mid P \mid M \,]\, V(T) \tag{3-52}$$

where

$$\dot{R}(T) \triangleq [\, \dot{\underline{R}}(T) \;\; \dot{\underline{R}}(2T) \cdots\cdots \dot{\underline{R}}[n_a(m+2)T]\,] \tag{3-53}$$

$$V(T) \triangleq [\, \underline{V}(T) \;\; \underline{V}(2T) \cdots\cdots \underline{V}[n_a(m+2)T]\,] \tag{3-54}$$

where

$$\dot{R}(T) \in R^{n_a \times n_a(m+2)}$$

$$V(T) \in R^{n_a(m+2)\times n_a(m+2)}$$

Main Result : Theorem 3-1 :

The input-output measurement data from a completely observable system may be used to determine its order providing the input and output vectors are continuously integrable. The proof of the theorem follows from the preceding development; the following summary with single-input-single-output highlights the main numerical features.

(a) $n = n_a$: assumed order is equal to the "true" order.

From equation (3-49) where we have

$$\dot{R}(t) = [\, A \mid P \mid M \,] \begin{bmatrix} \underline{R}(t) \\ \underline{F}(t) \\ \underline{G}(t) \end{bmatrix}$$

In expanded form

$$
\begin{bmatrix} \dot{y}_n(t) \\ \dot{y}_{n-1}(t) \\ \vdots \\ \dot{y}_2(t) \\ \dot{y}_1(t) \end{bmatrix} = \left[\begin{array}{cccccc|c|c} 0 & 1 & 0 & 0 & \cdots & 0 & & \\ 0 & 0 & 1 & 0 & \cdots & 0 & & \\ & & \vdots & & & & P & M \\ 0 & 0 & \cdots & & \cdots & 1 & & \\ -a_1 & -a_2 & \cdots & & -a_{n-1} & -a_n & & \end{array}\right] \begin{bmatrix} y_n(t) \\ y_{n-1}(t) \\ \vdots \\ y_2(t) \\ y_1(t) \\ \hline \underline{F}(t) \\ \hline \underline{G}(t) \end{bmatrix} \tag{3-55}
$$

where

$A \in R^{n \times n}$

It is clear from the structure of A that $\dot{y}_i(t) = y_{i-1}(t)$ or $y_i(t) = \int_{t_0}^{t} y_{i-1}(t)\,dt$ according to the definition of $y_i(t)$.

(b) $n_s < n$

Let $n_s = n-1$ i.e., $y(t)$ is integrated $(n-1)$ times thus from (3-55) we have

$$
\begin{bmatrix} \dot{y}_{n-1}(t) \\ \dot{y}_{n-2}(t) \\ \vdots \\ \dot{y}_1(t) \end{bmatrix} = \left[\begin{array}{ccccc|c|c} 0 & 1 & 0 & 0 & 0 & & \\ 0 & 0 & 1 \cdots & 0 & 0 & & \\ & & \vdots & & & P & M \\ -a_2 & -a_3 & \cdots & & -a_n & & \end{array}\right] \begin{bmatrix} y_{n-1}(t) \\ y_{n-2}(t) \\ \vdots \\ y_2(t) \\ y_1(t) \\ \hline \underline{F}(t) \\ \hline \underline{G}(t) \end{bmatrix} \tag{3-56}
$$

Note that $A \in R^{(n-1)\times(n-1)}$ in equation (3-55) has the canonical form as in equation (3—55) though the elements a_i $i=2,3,\cdots,n$ will have different values than those of a_i, also they are not unique for different sampling periods.

(c) $n_a > n$

Let $n_a = n+1$ i.e., $y(t)$ is integrated $(n+1)$ times, thus from (3—55) we have :

$$\begin{bmatrix} \dot{y}_{n+1}(t) \\ \dot{y}_n(t) \\ \vdots \\ \vdots \\ \dot{y}_2(t) \\ \dot{y}_1(t) \end{bmatrix} = \left[\begin{array}{ccccc|c|c} B_{11} & B_{12} & \cdots & B_{1n} & B_{1(n+1)} & & \\ 0 & 1 & \cdots & \cdots & B_{2(n+1)} & & \\ 0 & 0 & 1 & \cdots & 0 & P & M \\ \vdots & \vdots & & \vdots & \vdots & & \\ -a_1 & -a_2 & \cdots & -a_n & -a_{n+1} & & \end{array}\right] \left[\begin{array}{c} y_{n+1}(t) \\ y_n(t) \\ \vdots \\ y_1(t) \\ \hline \underline{F}(t) \\ \hline \underline{G}(t) \end{array}\right] \qquad (3\text{—}57)$$

Now from (3—57) it is clear that the relation $\dot{y}_i(t) = y_{i-1}(t)$ does not hold any more and the computed A matrix will have a completely different form from that in equation (3—55). Also the elements will not be unique for different sampling periods.

Thus if $n_a < n$, and if $V(T)$ is non-singular for $T = T_1$ then the estimated A will have the form of (3—3) for T_1, T_2, but the elements of the last p-rows will be different for different values of T.

If $n_a = n$, and if $I(T)$ is non-singular for $T = T_1$, $T = T_2$ then the estimated A will have the form (3—3) for T_1, T_2, and the elements of the last p-rows will be the same for different values of sampling intervals T.

3-2 Order determination procedure

The above results form the basis of determining the unknown order, the main stages of computation are listed below

Step 1 Consider the linear time invariant system $\overline{S}$ specified by equations (3-1) and (3-2)

$$\overline{S}: \begin{array}{l} \dot{\underline{x}}(t)=A\underline{x}(t)+B\underline{u}(t) \qquad ((3-1)) \\ \underline{y}(t)=C\underline{x}(t) \qquad ((3-2)) \end{array}$$

Step 2 Starting with an assumed order n_a say $n_a=2$, measure the inputs $\underline{u}(t)$, outputs $\underline{y}(t)$ and their successive integrals $\underline{u}_1(t)$, $\underline{u}_2(t)$, ... $\underline{u}_{n_a}(t)$ and $\underline{y}_1(t)$, $\underline{y}_2(t)$, ... $\underline{y}_{n_a}(t)$ at $n_a(m+2)$ equal interval of time, with a sampling interval T.

Step 3 From the measured data construct the matrices $\dot{R}(T)$ and $V(T)$ as defined in (3-53) and (3-54)

$$\dot{R}(T) \triangleq [\,\dot{\underline{R}}(T)\ \ \dot{\underline{R}}(2T) \cdots\cdots \dot{\underline{R}}[n_a(m+2)T]\,] \qquad ((3-53))$$

$$V(T) \triangleq [\,\underline{V}(T)\ \ \underline{V}(2T) \cdots\cdots \underline{V}[n_a(m+2)T]\,] \qquad ((3-54))$$

Step 4 Compute the composite matrix $[A \,\vdots\, P \,\vdots\, M]$ from (3-52)

$$\dot{R}(T) = [A \,\vdots\, P \,\vdots\, M]\, V(T) \qquad ((3-52))$$

$$[A \,\vdots\, P \,\vdots\, M] = \dot{R}(T)\,[V(T)]^{-1} \qquad (3-58)$$

Step 5 If the estimated matrix $\hat{A}$ is found to be in the special form defined in (3—3) then increase the assumed order by one and repeated step 2, it means $n_a = 3$.

Step 6 Repeat the above steps until the form of the estimated A is no longer in the form of (3—3). The actual system order then is $n = n_a - 1$, where n_a is the assume order at which the form of A has changed from the defined special form.

A general block diagram of the data processing stages for order determination is shown in Fig. 3-1. The physical system is external to the computer. The data capturing hardware is a purpose-built interfacing circuit compatible with the software and hardware configuration of the computer hosting the identification algorithm.

(a) Choice of n_a and T

The order determination procedure starts off with an initial value determined by the engineer (based on his judgement and experience). Since first order systems are relatively easy to identify from their step responses, in general the initial order may be assumed to be 2, i.e., initial value of n_a is 2.

As with the algorithm in Chapter-2, the choice of the sampling interval (T) for the digital data capturing circuit is not unique, but the designer needs to assign a value of T. The choice of T and its influence on estimation are discussed later, but for convenience, T is assumed to have a finite value in the procedure below.

(b) Collection of measurement data

With assigned values of n_a and T, the following time-

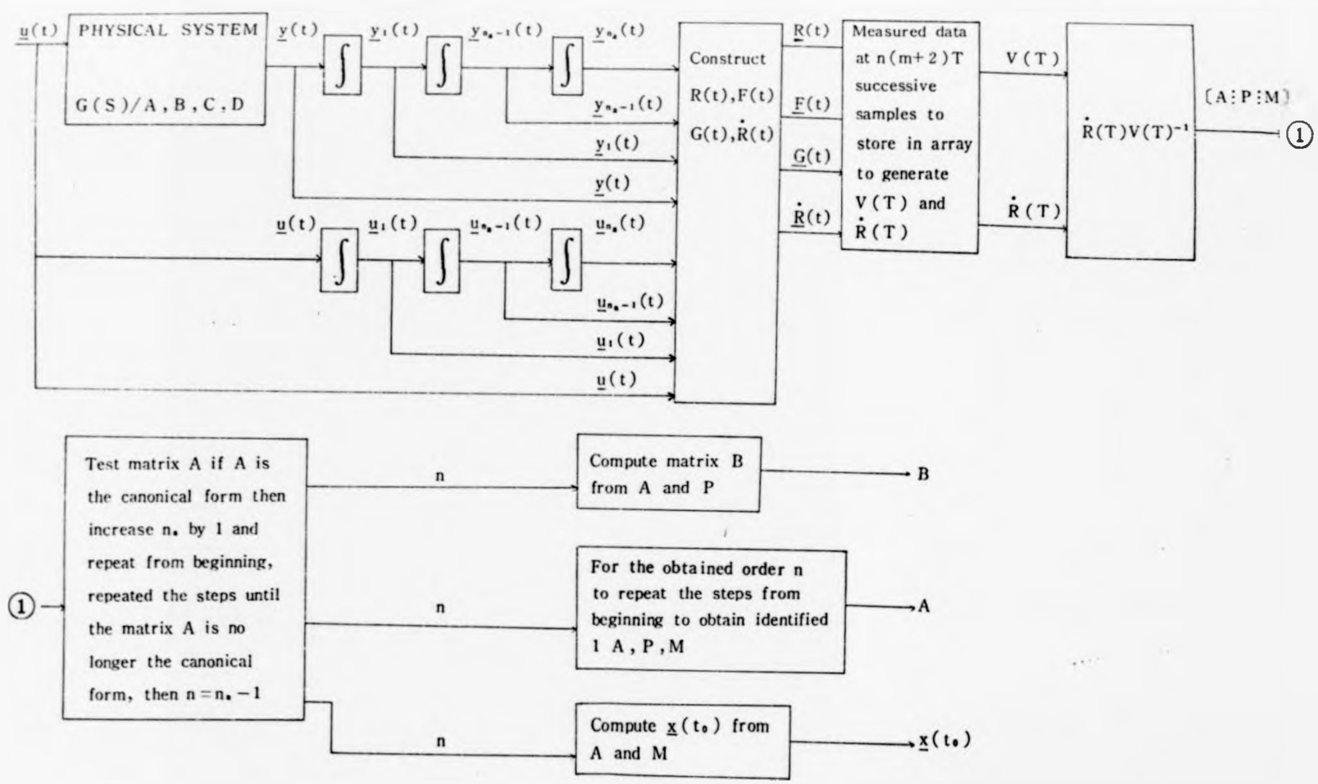

Fig. 3-1 The general block diagram of the order determination and parameter estimation

varying variables are sampled $n_s(m+2)$ times and stored :

input control vector $u(t)$

output response vector $y(t)$

(c) <u>Integration and computation with initial time t_0</u>

In addition to the above measurement data, the following integration and computation are performed and the results stored at each sampling interval

$$u_\alpha(t) = \underbrace{\int_{t_0}^{t} \cdots\cdots \int_{t_0}^{t}}_{\alpha} u(z)\,dz$$

$$y_\alpha(t) = \underbrace{\int_{t_0}^{t} \cdots\cdots \int_{t_0}^{t}}_{\alpha} y(z)\,dz$$

and

$$\frac{(t-t_0)^{\alpha-2}}{(\alpha-2)!} \qquad \text{for } \alpha = 1,\ 2,\ \cdots\cdots,\ n_s.$$

The sampled data collected from the above stages are :

$$\left.\begin{array}{l} \text{input vector } u(\beta T) \\ \text{output vector } y(\beta T) \\ \text{integrated input } u_\alpha(\beta T) \\ \text{integrated output } y_\alpha(\beta T) \\ \text{initial time } \dfrac{(\beta T - t_0)^{\alpha-2}}{(\alpha-2)!} \end{array}\right\} \text{ for } \beta = 1,\ 2,\ \cdots,\ [\,n_s(m+2)\,]$$

(d) From the above discrete data, the following real matrices are constructed ; for $\beta=1, 2, \cdots, n_s(m+2)$

$$F(\beta T)=\begin{bmatrix} u_{n_s}(\beta T) \\ \vdots \\ u_1(\beta T) \end{bmatrix}$$

$$E(\beta T)=\begin{bmatrix} y_{n_s}(\beta T) \\ \vdots \\ y_1(\beta T) \end{bmatrix}$$

and

$$G(\beta T)=\begin{bmatrix} \dfrac{(\beta T-t_0)^{n_s-1}}{(n_s-1)!} \\ \vdots \\ (t-t_0) \\ 1 \end{bmatrix}$$

In addition, the following matrix is constructed :

$$S=[\, I_{n\times n} \ \vdots\ O_{n\times(np-n)}\,]$$

using above and $E(\beta T)$, the matrix $R(\beta T)$ is formed as

$$R(\beta T)=SE(\beta T) \quad \text{for every } \beta\in 1, 2, \cdots, [n_s(m+2)]$$

The matrices $R(\beta T)$, $F(\beta T)$ and $G(\beta T)$ are ordered in the sequence below to form the $V(\beta T)$ matrix

$$V(\beta T)=\begin{bmatrix} R(\beta T) \\ F(\beta T) \\ \underline{G}(\beta T) \end{bmatrix} \quad \text{for each } \beta$$

Also from the integrated data in (c), the following matrices are constructed

$$\dot{G}(\beta T) = \begin{bmatrix} \dfrac{(\beta T - t_0)^{n_a-2}}{(n_a-2)!} \\ \vdots \\ 1 \\ 0 \end{bmatrix}$$

$$\dot{F}(\beta T) = \begin{bmatrix} u_{n_a-1}(\beta T) \\ \vdots \\ u(\beta T) \end{bmatrix}$$

$$\dot{R}(\beta T) = S\,\dot{E}(\beta T)$$

$$= S \begin{bmatrix} y_{n_a-1}(\beta T) \\ \vdots \\ y(\beta T) \end{bmatrix}$$

(e) The following composite matrices are now formed :

$$\dot{R}(T) = [\,\dot{R}(T) \quad \dot{R}(2T) \cdots\cdots \dot{R}\{n_a(m+2)T\}\,]$$

and

$$V(T) = [\,V(T) \quad V(2T) \cdots\cdots V\{n_a(m+2)T\}\,]$$

(f) The main order-determination algorithm is based on the relationship

$$\dot{R}(T) = [\,A \vdots P \vdots M\,]\,V(T)$$

when P and M are as defined in equations (3-46) and (3-47) and A is in canonical form.

The above equation is processed to obtain

$$\dot{R}(T)[\,V(T)\,]^{-1}$$

which by definition is equivalent to

$$[\, A \,\vdots\, P \,\vdots\, M \,]$$

From the computed result $\dot{R}(T)[V(T)]^{-1}$, the first $n_a \times n_a$ block is picked out and assigned a name A_a. The structure of A_a is now examined as below.

As indicated earlier for $n_a \leq n$, A_a is in canonical form but for $n_a > n$, A_a is not in canonical form. This observation forms the basis of the examination of A_a. The true value of the system order is that value of n_a beyond which A_a is not in canonical form. This " transition point " is the key feature of the order determination procedure.

Having derived A_a above, its structure is examined to check if it is in canonical form. If this is in canonical form, then n_a is increased by 1 and the whole procedure is repeated with this increased value of n_a. Let n_{af} be the value of n_a at which A_a becomes " non-canonical" for the first time, i.e., for $n_a = n_{af} - 1$ A_a is in canonical form and for $n_a = n_{af}$, A_a is not in canonical form. The true order of the system is then $n = n_{af} - 1$.

The above methodology forms the basis of the numerical algorithm developed to determine the order of a system from measurement data. Flow-chart of the program is shown in Fig. 3-2 and a complete listing is given in the Appendix (Program B). Two simple numerical examples are given below to illustrate the general validity of the above procedure.

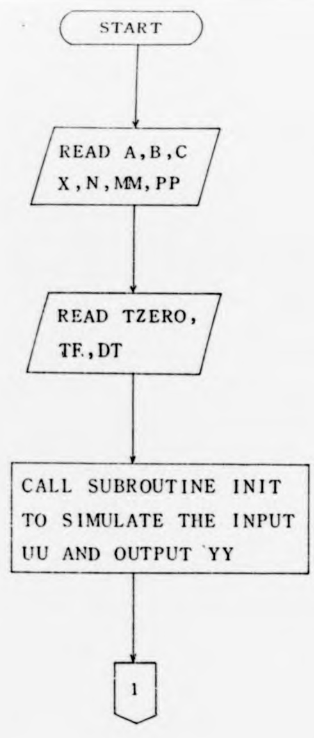

Fig. 3-2(1)

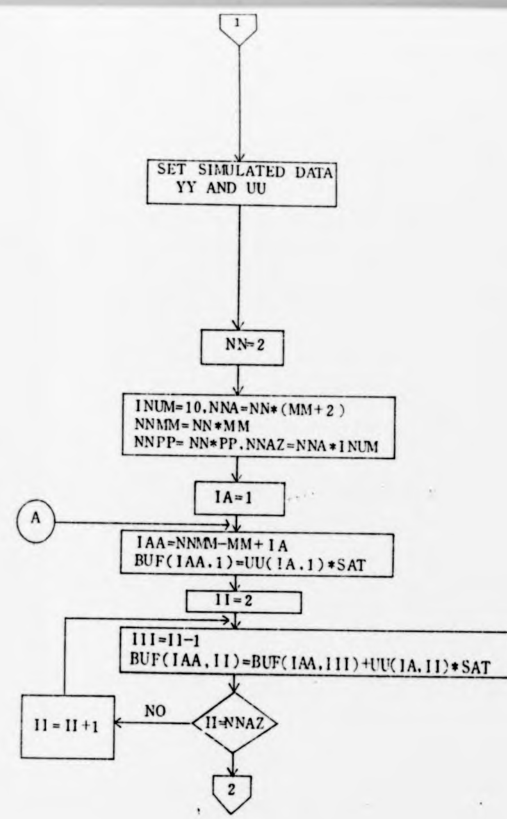

Fig. 3-2(2)

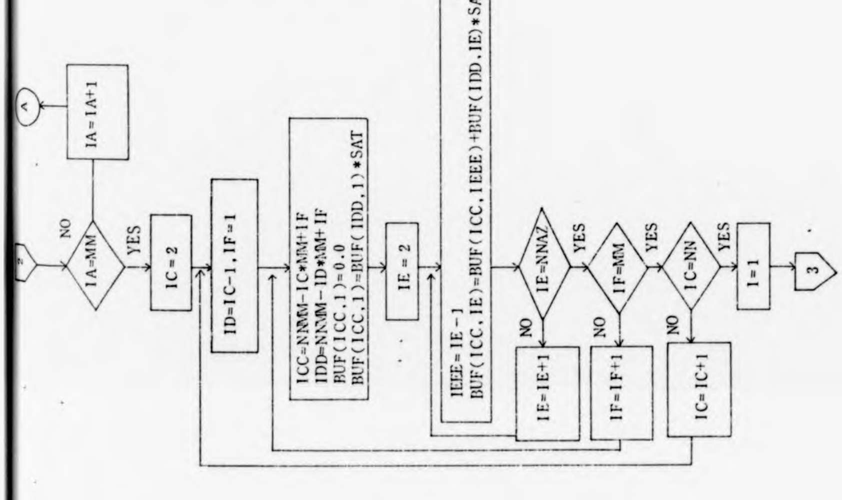

Fig. 3-2(3)

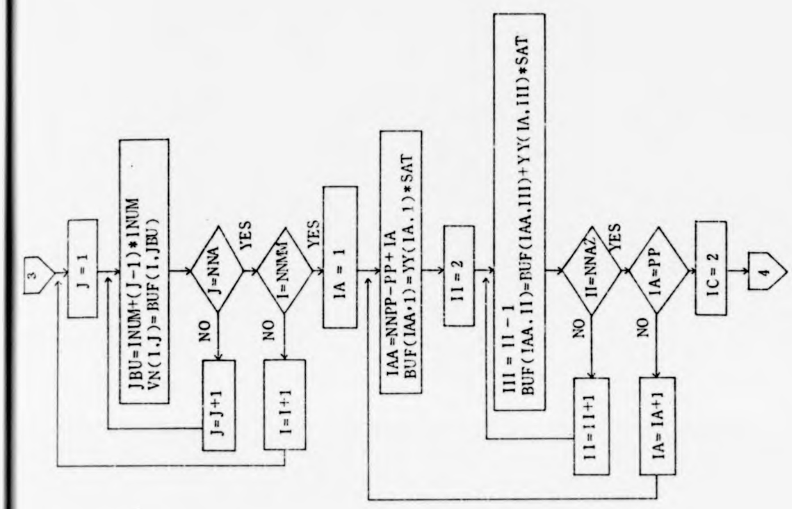

Fig. 3-2(4)

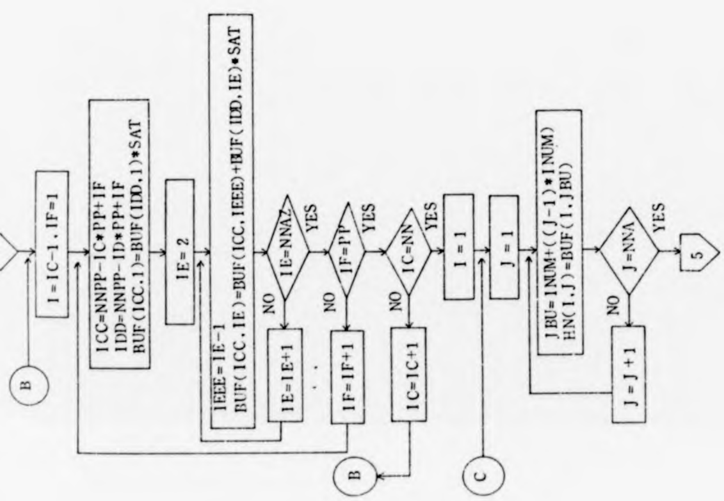

Fig. 3-2(5)

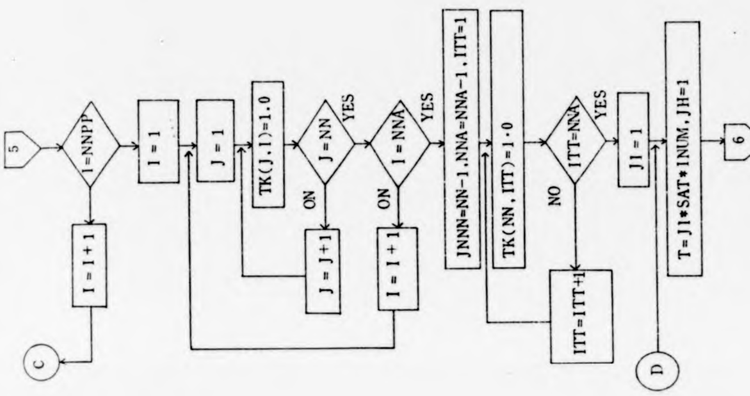

Fig. 3-2(6)

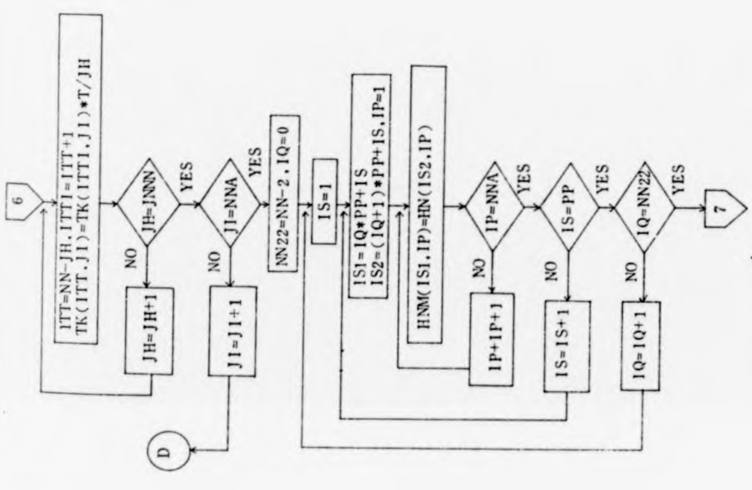

Fig. 3-2 (7)

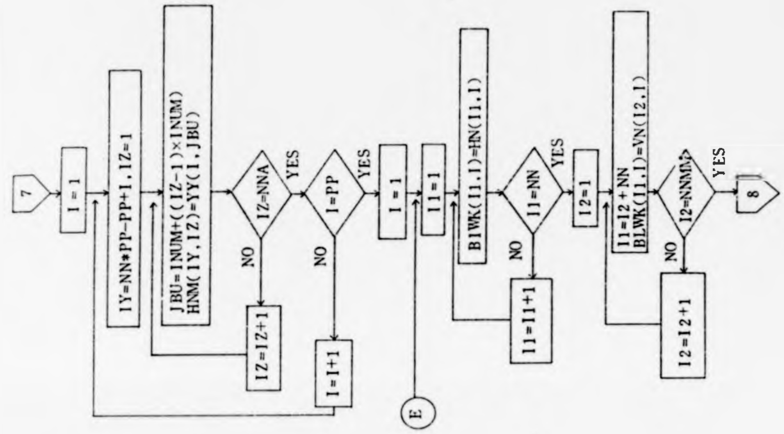

Fig. 3-2 (8)

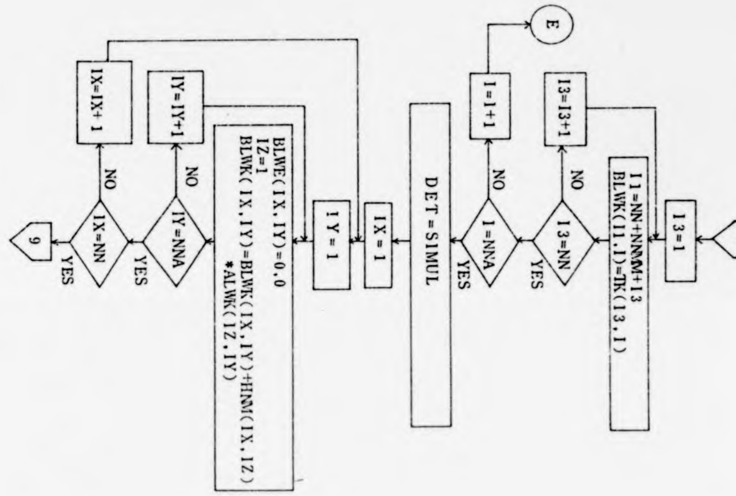

Fig. 3-2 (9)

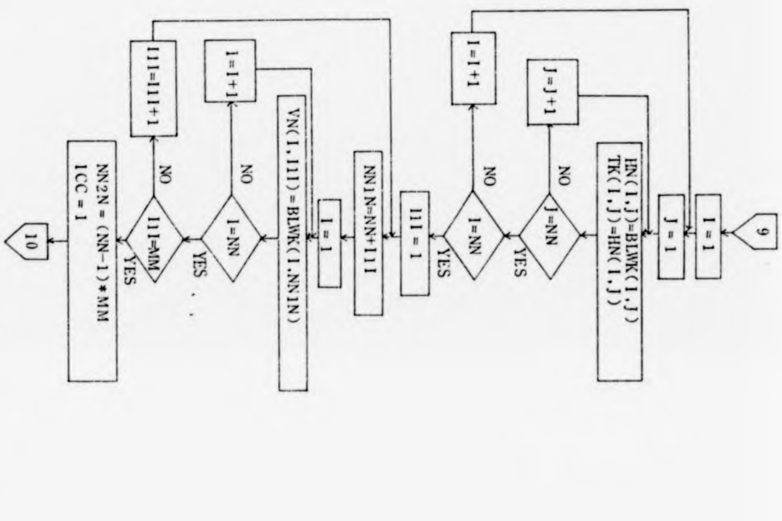

Fig. 3-2 (10)

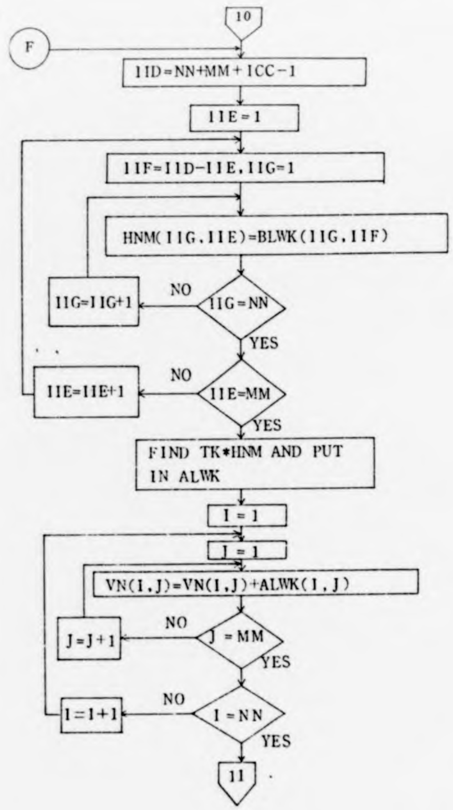

Fig. 3−2 ⑪

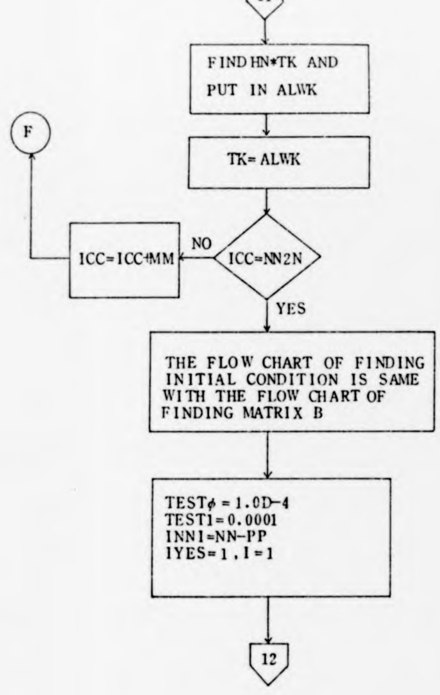

Fig. 3−2 ⑫

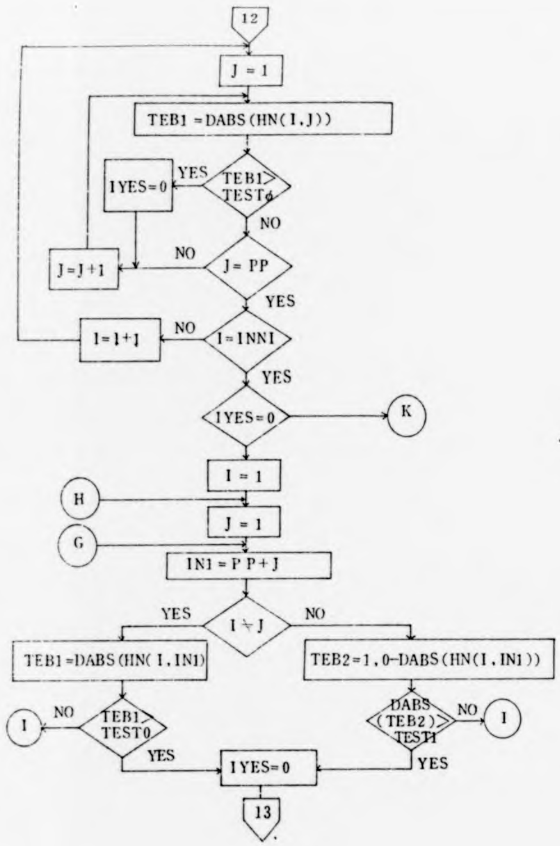

Fig. 3-2 (3)

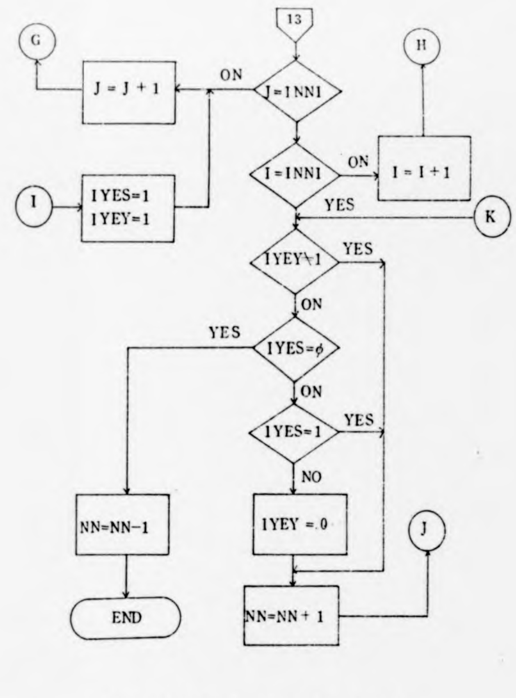

Fig.3-2 The flow chart of program B

Example 3 — 1 The " true system " is represented by

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ 1.0 & -2.0 & 3.0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 1.0 \\ 1.0 \\ 2.0 \end{bmatrix} u(t)$$

$$y(t) = [\, 1.0 \quad 0.0 \quad 0.0 \,] \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

$$\underline{x}(t_0) = \begin{bmatrix} 1.0 \\ 1.0 \\ 2.4 \end{bmatrix}$$

Using the biased sinusoidal input of the form

$u(t) = 1.0 + \sin(1.2t)$, and

assuming $n_s = 2$, the measurement data and integrated data are collected for $T = 30$ mseconds. These are used to construct the matrices:

$\dot{R}(T)$ and $V(T)$

These two matrices are " processed " to derive

$\dot{R}(T)\,[\,V(T)\,]^{-1}$

For the example, using a completely simulated true system and data capturing hardware as in Fig. 3 — 1 , the $\dot{R}(T)\,[\,V(T)\,]^{-1}$ is

$$\dot{R}(T)\,[\,V(T)\,]^{-1} =$$

$$
\left[\begin{array}{cc}
\overbrace{\begin{array}{cc} -1.164153\,D-9 & 1.000000\,D+00 \\ 3.336198\,D+00 & 3.556161\,D-01 \end{array}}^{A_1} \\
\overbrace{\begin{array}{cc} 0.000000\,D+00 & -1.358199\,D+00 \\ -6.257324\,D-10 & 1.202605\,D+00 \end{array}}^{P} \\
\overbrace{\begin{array}{cc} -3.346941\,D-10 & 1.230757\,D+00 \\ 4.718448\,D-16 & 9.916152\,D-01 \end{array}}^{M}
\end{array}\right]
$$

As can be seen, $n_a \times n_a$ square submatrix formed out of the first n_a columns of the above matrix is in canonical form. So repeatition of the numerical algorithm is needed. For $n_a = 3$, the procedure yields

$$
\dot{R}(T)\,[\,V(T)\,]^{-1} =
\left[\begin{array}{c}
\overbrace{\begin{array}{ccc} 1.811981\,D-05 & 9.999285\,D-01 & 3.147125\,D-05 \\ 4.110336\,D-04 & -6.256104\,D-04 & 1.000381\,D+00 \\ 9.988861\,D-01 & -1.997253\,D+00 & 2.996979\,D+00 \end{array}}^{A_a} \\
\overbrace{\begin{array}{ccc} 9.536743\,D-05 & 8.239746\,D-04 & 1.005371\,D+00 \\ -3.433228\,D-05 & -1.087189\,D-04 & -1.996689\,D+00 \\ 1.853704\,D-05 & 2.164841\,D-04 & 1.001007\,D+00 \end{array}}^{P} \\
\overbrace{\begin{array}{ccc} 6.794930\,D-06 & 7.152557\,D-05 & -9.994268\,D-01 \\ -4.768372\,D-01 & 4.272461\,D-04 & -1.186096\,D+00 \\ -4.597017\,D-17 & 2.636780\,D-16 & 9.916153\,D-01 \end{array}}^{M}
\end{array}\right]
$$

which shows that A_2 is in canonical form. So the procedure is repeated with $n_2 = 4$, this yields

$$
R(T)\,[\,V(T)\,]^{-1} = \left[\;\overbrace{\begin{array}{cccc}
2.685547\,D-03 & 9.792480\,D-01 & -4.785156\,D-02 & 1.123047\,D-02 \\
1.867676\,D-01 & -5.039062\,D-01 & 1.496094\,D+00 & -9.399414\,D-02 \\
3.503906\,D+00 & -8.996094\,D+00 & 8.933594\,D+00 & -2.964844\,D+00 \\
5.006250\,D+01 & -9.175000\,D+01 & 1.158750\,D+02 & -3.200000\,D+01
\end{array}}^{A_2}\right.\;\Bigg|
$$

$$
\overbrace{\begin{array}{cccc}
-1.464844\,D-03 & 1.547852\,D-01 & 2.667969\,D+00 & 4.025000\,D+01 \\
5.883789\,D-02 & 7.148437\,D-01 & 7.562500\,D+00 & 2.356250\,D+01 \\
-1.684570\,D-02 & -2.441406\,D-04 & 3.242188\,D+00 & 3.987500\,D+01 \\
1.220703\,D-03 & 1.315918\,D-01 & 3.230469\,D+00 & 1.500000\,D+01
\end{array}}^{P}\;\Bigg|
$$

$$
\left.\overbrace{\begin{array}{cccc}
-2.221680\,D-02 & -4.609375\,D-01 & -9.312500\,D+00 & -7.675000\,D+01 \\
-4.858398\,D-02 & -9.726562\,D-01 & -1.321484\,D+01 & -1.000625\,D+02 \\
-1.416016\,D-02 & -8.203125\,D-02 & 2.031250\,D-01 & 8.562500\,D+00 \\
6.938894\,D-18 & 4.857226\,D-16 & 7.577272\,D-15 & 9.916153\,D-01
\end{array}}^{M}\;\right]
$$

which shows that A_a is not in canonical form , hence the order of the true system is

$$n = n_{af} - 1 = 3$$

Example 3 − 2 The " true system " is represented by

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ -6.0000 & -11.0000 & -6.0000 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 1.0000 \\ 1.0000 \\ 2.0000 \end{bmatrix} u(t)$$

$$y(t) = [\, 1.0000 \quad 0.0000 \quad 0.0000 \,] \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

$$\underline{X}(t_o) = \begin{bmatrix} 1.3000 \\ 2.6000 \\ 4.5000 \end{bmatrix}$$

Using the biased sinusoidal input of the form

$u(t) = 1.0 + \sin(1.2\,t)$, and

assuming $n_s = 2$, the measurement data and integrated data are collected for $T = 30$ mseconds. These are used to construct the matrices :

$\dot{R}(T)$ and $V(T)$

These two matrices are " processed " to derive

$$\dot{R}(T)[\,V(T)\,]^{-1}$$

For the example, using a completely simulated true system and data capturing hardware as in Fig. 3-1, the $\dot{R}(T)[\,V(T)\,]^{-1}$ is

$$\dot{R}(T)[\,V(T)\,]^{-1} = \left[\begin{matrix} 2.328306\text{D}-09 & -1.319313\text{D}+02 \\ 1.000000\text{D}+00 & -1.777207\text{D}+01 \end{matrix}\right.$$

$$\begin{matrix} -6.752089\text{D}-09 & 2.762981\text{D}+02 \\ 1.076842\text{D}-09 & -1.666236\text{D}+01 \end{matrix}$$

$$\left.\begin{matrix} -9.167707\text{D}-10 & 4.304409\text{D}+01 \\ 1.360023\text{D}-15 & 1.289215\text{D}+00 \end{matrix}\right]$$

As can be seen, $n_s \times n_s$ square submatrix formed out of the first n_s columns of the above matrix is in canonical form. So repeatition of the numerical algorithm is needed. For $n_s = 3$ given

$$\dot{R}(T)[\,V(T)\,]^{-1} = \left[\begin{matrix} 1.782179\text{D}-05 & 1.000003\text{D}+00 & 2.980232\text{D}-06 \\ 1.754761\text{D}-04 & 2.288818\text{D}-05 & 1.000005\text{D}+00 \\ -6.052231\text{D}+00 & -1.107199\text{D}+01 & -6.020921\text{D}+00 \end{matrix}\right.$$

$$\begin{matrix} -2.384186\text{D}-05 & -3.356934\text{D}-04 & 1.917232\text{D}+01 \\ 3.159046\text{D}-06 & 7.569790\text{D}-06 & 7.006070\text{D}+00 \\ -1.031160\text{D}-05 & -1.115799\text{D}-04 & 9.976959\text{D}-01 \end{matrix}$$

$$
\left.\begin{array}{rrr}
5.722046\text{D}-06 & 7.152557\text{D}-05 & 3.466612\text{D}+01 \\
7.748604\text{D}-06 & 8.201599\text{D}-05 & 1.039162\text{D}+01 \\
-5.412337\text{D}-16 & -3.941292\text{D}-15 & 1.289225\text{D}+00
\end{array}\right]
$$

which shows that A_s is in canonical form. So the procedure is repeated with $n_s = 4$, this yields

$$
R(T)[V(T)]^{-1} = \left[\begin{array}{rrrr}
1.149902\text{D}-01 & 1.132813\text{D}+00 & 9.057617\text{D}-07 & 6.591797\text{D}-03 \\
-4.101562\text{D}-01 & -2.234375\text{D}+00 & 3.984375\text{D}-01 & -1.665039\text{D}-01 \\
-2.250000\text{D}+00 & -1.300000\text{D}+01 & -3.062500\text{D}+00 & -1.132812\text{D}-01 \\
-6.062500\text{D}+00 & -3.843750\text{D}+01 & -2.043750\text{D}+01 & -1.043750\text{D}+01
\end{array}\right.
$$

$$
\begin{array}{rrrr}
-3.242187\text{D}-01 & 1.007813\text{D}+00 & 5.875000\text{D}+00 & -2.100000\text{D}+01 \\
-1.049805\text{D}-02 & 2.796875\text{D}+00 & 3.062500\text{D}+01 & 1.901250\text{D}+02 \\
-3.662109\text{D}-03 & 1.743164\text{D}-01 & 1.710938\text{D}+00 & 1.656250\text{D}+01 \\
3.204346\text{D}-03 & 6.520996\text{D}-01 & 5.672852\text{D}+00 & 4.631250\text{D}+01
\end{array}
$$

$$
\left.\begin{array}{rrrr}
-5.703125\text{D}-01 & -1.500000\text{D}+00 & -2.325000\text{D}+01 & -1.550000\text{D}+02 \\
-2.856445\text{D}-02 & 2.273438\text{D}+00 & 8.312500\text{D}+00 & 4.168750\text{D}+01 \\
-2.539062\text{D}-02 & -4.765625\text{D}-01 & -5.843750\text{D}+00 & -3.587500\text{D}+01 \\
-1.561251\text{D}-17 & 1.207368\text{D}-15 & 8.423817\text{D}-15 & 1.289225\text{D}+00
\end{array}\right]
$$

which shows that A_s is not in canonical form, hence the order of the true system is

$$n = n_{s,f} - 1 = 3$$

3—3 Parameter determination

The mathematical derivation in Section 3—1 and the associated results presented as Theorem 3—1, indicate that the inherent properties between the input control and output response and their time integrations may be used for the estimation of the dynamics matrix A and the input matrix B, providing the output matrix C is expressed in the special form shown in equation (3—4). In view of this, it is worth exploring the above mathematical/structural properties to derive a suitable method for the computation of A and B as well as the system order from the measurement data within a single mathematical framework.

For notational convenience the following vectors are defined:

$$\tilde{y}_{\alpha}(t) = \begin{bmatrix} \underbrace{\int_{t_0}^{t} \cdots\cdots \int_{t_0}^{t}}_{\alpha} y_1(t)\,dt \\ \vdots \\ \underbrace{\int_{t_0}^{t} \cdots\cdots \int_{t_0}^{t}}_{\alpha} y_p(t)\,dt \end{bmatrix} \rightarrow \begin{array}{l} \alpha \text{ th integration of} \\ \text{the output vector} \\ \tilde{y}(t) \end{array}$$

the notatian $\sim$ indicates that it is a vector, without this notation e.g., $y_1(t)$ is the first integration of the scalar output $y(t)$. Similarly,

$$\widetilde{u}_\alpha(t) = \begin{bmatrix} \underbrace{\int_{t_0}^{t} \cdots\cdots \int_{t_0}^{t}}_{\alpha} u_1(t)\,dt \\ \vdots \\ \underbrace{\int_{t_0}^{t} \cdots\cdots \int_{t_0}^{t}}_{\alpha} u_m(t)\,dt \end{bmatrix}$$

and

$$\widetilde{x}_\alpha(t) = \begin{bmatrix} \underbrace{\int_{t_0}^{t} \cdots\cdots \int_{t_0}^{t}}_{\alpha} x_1(t)\,dt \\ \vdots \\ \underbrace{\int_{t_0}^{t} \cdots\cdots \int_{t_0}^{t}}_{\alpha} x_n(t)\,dt \end{bmatrix}$$

with these notations, the main equation of the previous section may be written as (in continuous form) :

$$\dot{R}(t) = [\, A_* \mid P \mid M \,] \begin{bmatrix} R(t) \\ F(t) \\ G(t) \end{bmatrix}$$

where A_* coresponds to a matrix for a given n_* which is identical to the true matrix A if $n_* = n$ (the order of the true system) .

In expanded form :

$$\begin{bmatrix} \dot{\tilde{y}}_n(t) \\ \dot{\tilde{y}}_{n-1}(t) \\ \vdots \\ \dot{\tilde{y}}_2(t) \\ \dot{\tilde{y}}_1(t) \end{bmatrix} = \left[\begin{array}{ccccc|c|c} 0 & 1 & 0 & 0 \cdots\cdots & 0 & & \\ 0 & 0 & 1 & 0 \cdots\cdots & 0 & & \\ & & & & & P & M \\ 0 & 0 & \cdots\cdots & \cdots\cdots & 1 & & \\ -a_1 & -a_2 \cdots & -a_{n-1} & & -a_n & & \end{array}\right] \left[\begin{array}{c} \tilde{y}_n(t) \\ \tilde{y}_{n-1}(t) \\ \vdots \\ \tilde{y}_2(t) \\ \tilde{y}_1(t) \\ \hline \underline{F}(t) \\ \hline \underline{G}(t) \end{array}\right] \qquad ((3\text{–}55))$$

where

$A \in R^{n \times n}$

It is clear from the structure of A that $\dot{\tilde{y}}_i(t) = \tilde{y}_{i-1}(t)$ or $\tilde{y}_i(t) = \int_{t_0}^{t} \tilde{y}_{i-1}(t)\, dt$ according to the definition of $\tilde{y}_i(t)$.

Now assume that the assumed order n_a is less than the actual order n suppose that $n_a = n-1$, i.e., y(t) is integrated (n−1) times thus from (3−55)

we have

$$\begin{bmatrix} \dot{\tilde{y}}_{n-1}(t) \\ \dot{\tilde{y}}_{n-2}(t) \\ \vdots \\ \vdots \\ \dot{\tilde{y}}_1(t) \end{bmatrix} = \left[\begin{array}{ccccc|c|c} 0 & 1 & 0 \cdots\cdots & 0 & 0 & & \\ 0 & 0 & 1 \cdots\cdots & 0 & 0 & & \\ & & \vdots & & & P & M \\ & & \vdots & & & & \\ -a_2 & -a_3 & \cdots\cdots & \cdots & -a_n & & \end{array}\right] \left[\begin{array}{c} \tilde{y}_{n-1}(t) \\ \tilde{y}_{n-2}(t) \\ \vdots \\ \tilde{y}_2(t) \\ \tilde{y}_1(t) \\ \hline \underline{F}(t) \\ \hline \underline{G}(t) \end{array}\right] \qquad ((3\text{–}56))$$

Note that $A_* \in R^{(n-1)\times(n-1)}$ still has the same canonical form as in equation (3—55), though the elements a_i $i=2,3,\cdots,n$ will have different values than those of a_i, but they are not unique for different sampling intervals.

Let the assumed order n_* is higher than the actual order n. Suppose that $n_* = n+1$, then, $\tilde{y}(t)$ is integrated $(n+1)$ times, thus from (3—55) we have:

$$\begin{bmatrix} \dot{\tilde{y}}_{n+1}(t) \\ \dot{\tilde{y}}_n(t) \\ \vdots \\ \dot{\tilde{y}}_2(t) \\ \dot{\tilde{y}}_1(t) \end{bmatrix} = \left[\begin{array}{ccccc|c|c} B_{11} & B_{12} & \cdots & B_{1n} & B_{1(n+1)} & & \\ 0 & 1 & \cdots & \cdots & B_{2(n+1)} & & \\ 0 & 0 & 1 & \cdots & 0 & P & M \\ \vdots & & & & & & \\ -a_1 & -a_2 & \cdots & -a_n & -a_{n+1} & & \end{array}\right] \left[\begin{array}{c} \dot{\tilde{y}}_{n+1}(t) \\ \dot{\tilde{y}}_n(t) \\ \vdots \\ \dot{\tilde{y}}_1(t) \\ \hline \underline{F}(t) \\ \hline \underline{G}(t) \end{array}\right]$$

((3—57))

Now from (3—55) it is clear that the relation $\dot{\tilde{y}}_i(t) = \dot{\tilde{y}}_{i-1}(t)$ does not hold any more and the A_* matrix will have a completely different form from that in equation (3—55). Also the elements are not be unique for different sampling periods.

So now if $n_* < n$, and if $V(T)$ is non-singular for $T=T_1$, $T=T_2$ then the matrix A_* will have the form of (3—3) for T_1, T_2, but the elements of the last p-rows will be different for different values of T.

If $n_* = n$, and if $V(T)$ is non-singular for $T=T_1$, $T=T_2$ then the matrix A will have the form of (3—3) for T_1, T_2, and the elements of the last p-rows will be the same for different values of T.

These observations permit the determination of n, the order of the system to be identified.

For this value of order n, we can continue to identify the system parameters.

(a) Computation of the dynamics matrix A

As seen earlier (Theorem 3-1) for $n_* < n$, if the true system is completely observable, the V(T) is non-linear and $\dot{R}(T)[V(T)]^{-1}$ can be evaluated and the submatrix A_* may be formed from the first n_* columns of the derived composite matrix. By increasing the assumed order, the "transition point" where A_* changes from canonical to non-canonical form, the order may to derived. Once the order is determined, then parameter identification can proceed.

The main observation which forms the basis of the estimation of A and B matrices is that for n_* = true order n, the elements of A_* in the matrix product.

$\dot{R}(T)[V(T)]^{-1}$

are unique for any given value of the sampling interval T. Thus when the "transition point" has been reached, the submatrix A_* may be used as an estimation of the true matrix A. To introduce a certain amount of reliability in this procedure, the above stages may be repeated with $n_e = n_f - 1 = n$ and a different set of sampling intervals T_1 and T_2. If $A_e \simeq A$ within the acceptable error-band, then

$$A_e(T_1) \simeq A_e(T_2) \simeq A$$

and this would conclude the estimation of the true dynamics matrix A.

(b) Computation of the input matrix B

Having obtained the dynamics matrix A_*, the above derivations may be extended to estimate the input matrix B. The following derivations illustrate the method.

From equation (3—46) we have

$$P = SK - ASJ$$

$$= [\, P_0 \ \ P_1 \ \ P_2 \cdots\cdots P_{n-1} \,] \qquad (3\text{—}59)$$

where

$$P_i = SQ_iB - ASQ_{i+1}B \qquad (3\text{—}60)$$

$$P_i \in R^{n\times m}$$

$$P_{n-1} = SQ_{n-1}B - ASQ_nB$$

since

$$Q_n = 0$$

then

$$P_{n-1} = SQ_{n-1}B$$

$$P_{n-2} = SQ_{n-2}B - ASQ_{n-1}B$$

$$= SQ_{n-2}B - AP_{n-1}$$

$$P_{n-3} = SQ_{n-3}B - ASQ_{n-2}B$$

$$= SQ_{n-3}B - AP_{n-2} - A^2P_{n-1}$$

$$\vdots$$

$$P_0 = -AP_1 - A^2P_2 - \cdots\cdots - A^{n-1}P_{n-1} + SQ_0B$$

Note that

$$SQ_0 = I$$

then

$$B = P_0 + AP_1 + A^2P_2 + \cdots\cdots + A^{n-1}P_{n-1} \qquad (3\text{—}61)$$

Thus with the known dynamics matrix $A = A_n$, the $n_n+1, \cdots\cdots, 2n_n$ columns of the composite matrix $\dot{R}(T)[V(T)]^{-1}$ may be picked out to derive an estimation of the input matrix, since

$$\dot{R}(T)[V(T)]^{-1} = [A_n \vdots \underbrace{P_0 P_1 \cdots\cdots P_{n_n-1}}_{P} \vdots M]$$

This gives

$$B_n = \sum_{i=0}^{n_n-1} A_i P_i \quad \text{with} \quad n_n = n$$

(c) Computation of the initial state vector $\underline{x}(t_o)$

With estimated matrices, A_s and B_s, the initial state of the true system may be computed:

$$M = SW - ASH = [M_0 \ M_1 \cdots\cdots M_{n-1}] \tag{3-62}$$

where

$$M_i = SQ_i\underline{x}(t_0) - ASQ_{i+1}\underline{x}(t_0) \tag{3-63}$$

$$M_i \in R^{n\times 1}$$

In the same manner, we get

$$\underline{x}(t_0) = M_0 + AM_1 + \cdots\cdots + A^{n-1}M_{n-1} \tag{3-64}$$

where the submatrice $M_0, M_1, \cdots, M_{n-1}$ are pick out from the derived matrix

$$\dot{R}(T)[V(T)]^{-1} = [A_s \vdots P \vdots M_o M_1 \cdots\cdots\cdots M_{n_s-1}]$$

The above results form the basis of the following parameter identification as an alternative to the procedure developed in the previous chapter. The main advantage of this algorithm is that it determines the order as well as the system parameters using the same set of input and output measurement data. Here the state vector need not be accessible — this makes this alternative method particularly suitable for many physical systems.

3 — 4 Identification algorithm and illustrative examples

The algorithm form of the above identification method is described below.

Step 1 Starting with an assumed order say $n_a = 2$ measure the inputs, outputs and their successive integrals at $n_a(m+2)$ equal samples of time, with a sampling interval T.

Step 2 From the measured data construct the matrices $\dot{R}(T)$ and $V(T)$ as defined in (3-53) and (3-54).

Step 3 From equation (3-52) compute the composite matrix $\dot{R}(T)[V(T)]^{-1}$ which by definition consiste three blocks $[A \mid P \mid M]$.

Step 4 If the matrix formed out of the first n_a columns (A_a) is found to be in the special form defined in equation (3-3), then increase the assumed order by one and repeat Step 2.

Step 5 Repeat the above steps until the structure of the computed matrix A_a is no longer in the form of equation (3-3). The actual system order then is $n = n_a - 1$, where n_a is the assumed order at which the form of A_a has changed from the defined special form.

Step 6 For the obtained order (n_a), choose a different sampling period and compute the matrix $[A \vdots P \vdots M]$.

Step 7 If the computed composite matrix $\dot{R}(T)[V(T)]^{-1}$ is the same for the two sampling intervals, then the solution is unique and the computed matrices contain information about the actual parameters of the given system.

Step 8 Compute the input matrix B and the initial states $\underline{x}(t_0)$ from equation (3—12) and (3—13) respectively.

The flow-chart of computer program is same Fig. 3－2, and listing (Program B) is as given similar to in Appendix 2. and some illustrative examples are given below.

Example 3 − 3 The system considered here to demonstrate the validity of the above algorithm has the following transfer function

$$\begin{bmatrix} y_1(s) \\ \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{s+1}{s^2+s+1} & \dfrac{-1}{s^2+s+1} \\ \dfrac{1}{s^2+s+1} & \dfrac{s}{s^2+s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ \\ u_2(s) \end{bmatrix}$$

In this example the system is treated as the physical system of unknown order and unknown parameters. To distinguish the above " physical system " (simulated for the illustration here) from the " identified system " the above system is referred to here as the " true system " as opposed to the (approximate) identified system. The state-space representation of the true system is

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0.0 & 1.0 \\ -1.0 & -1.0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

$$
\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}
$$

The general identification configuration is shown in Fig. 3-1. where sinusoidal input signals are used to generate the output response needed for the identification algorithm. For this illustrative example, the physical system as well as the identification hardware were simulated on the digital computer. The data from the " hardware " were used as the input measured data into the identification software. For an arbitrary choice of $T = 0.1$ msecond, the identified parameters generated by the software are

$$
\hat{A} = \begin{bmatrix} 7.17163\text{D}-04 & 1.0005020\text{D}+00 \\ -1.000366\text{D}+00 & -9.999676\text{D}-01 \end{bmatrix}
$$

$$
\hat{B} = \begin{bmatrix} 1.000031\text{D}+00 & -5.999143\text{D}-04 \\ -4.864478\text{D}-04 & 1.000432\text{D}+00 \end{bmatrix}
$$

$$
\hat{\underline{x}}(t_o) = \begin{bmatrix} -1.061440\text{D}-03 \\ -2.532959\text{D}-03 \end{bmatrix}
$$

Although fairly good agreement is between the true parameters and the identified parameter is obtained in this algorithm, the extent of error introduced in the identification procedure is very much dependent on the choice of T, the sampling interval. In general, the Nyquist sampling theorem could be used here to choose T. However, in a completely automated identification scheme, the natural frequency or the waveforms of the measurement continuous signals may not be known. Although several established identification algorithms use very small sampling intervals, there is no established guidelines on the choice of T(apart from the hardware/measurement constraints imposed by a particular interface circuitry). Some of the aspects associated with the choice of T, in relation to the transient behaviour of the system being identified, are considered through the following examples.

Example 3 – 4 overdamped system : The true system is represented by

$$G(s) = \frac{2.5}{s^2 + 11s + 10}$$

for which

$$A = \begin{bmatrix} 0.0 & 1.0 \\ -10.0 & -11.0 \end{bmatrix} , \quad B = \begin{bmatrix} 0.0 \\ 2.5 \end{bmatrix}$$

$$C = [\ 1.0 \quad 0.0\ ] \quad , \quad \underline{x}\,(t_o) = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$$

which suggest that the dominant time constant T_c of the system is 1 second. The biased sinusoidal input signals is used to generate the output response needed for the identification algorithm. Consequently, the identification algorithm developed earlier is applicable. Applying the procedure indicated by Fig. 3 – 1. the choice of sampling interval $T = 0.10$m seconds ($T_c / 10{,}000$) gives system the following estimation for the system parameters :

$$\hat{A} = \begin{bmatrix} 2.235174\text{D}-08 & 1.000000\text{D}+00 \\ -1.000601\text{D}+01 & -1.100555\text{D}+01 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} -1.361386\,D-04 \\ 2.503009\,D+00 \end{bmatrix}$$

$$\underline{\hat{x}}\,(t_o) = \begin{bmatrix} 1.489827\,D-08 \\ -2.773281\,D-04 \end{bmatrix}$$

The estimation errors (2-48) and (3-71) for this choice of T are :

$$S_A = 0.1673\,D-04$$

$$S_B = 0.4536\,D-05$$

$$S_O = 0.3846\,D-07$$

$$S = 0.7102\,D-05$$

which are acceptable for most analysis/design studies using linear theory.

To study the effect of the choice of the sampling interval (T) on the estimation errors (S_A, S_B, S_O, S), T was varied from 0.03 mseconds to 0.23 mseconds, the resulting errors are ploted in Fig. 3-3. The following observation can be made from these error curves :

(1) square of the estimation errors are small in amplitude and well within the bound acceptable for most design procedures over a wide range sampling intervals (T=0.03

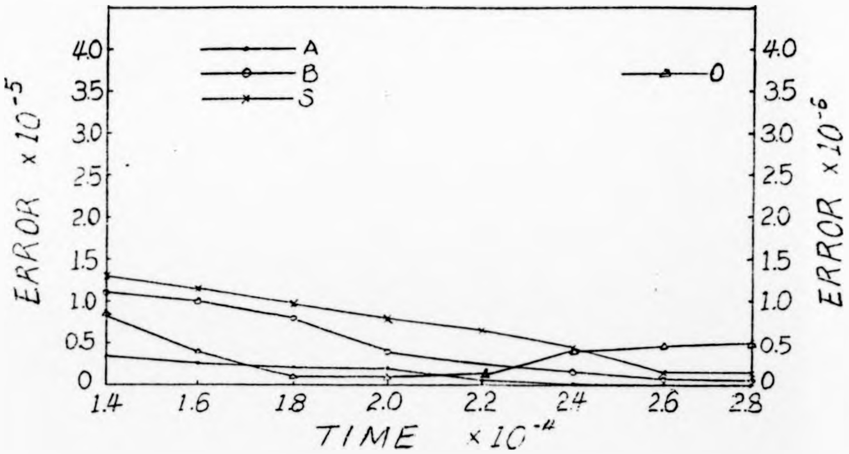

Fig. 3 — 3 The effect of varying T on the estimation errors (overdamped)

mseconds to 0.23 mseconds). For the system in Example 3 — 4 , the "optimum" choice is arourd 0.10 mseconds which is 10^{-4} times of the dominant time constant of the system ($T_c = 1$ second) .

(2) It is interesting to observe that the estimation errors are higher at the two boundaries. The estimation algorithm is based the mathematical theorem which constructs the two matrices V(t) (3 — 50) and $\dot{R}(t)$ (3 — 51) out of the n(m+2)T samples of each parameter $y(t)$, $y_1(t) = \int y_1(t)dt$, $y_2(t) = \int y_1(t)dt$, $u(t)$, $u_1(t) = \int u(t)dt$ and $(t - t_o)$, the mathematics of the algorithm is based on processing these 5 sets of n(m+2)T discrete data

and consequently, the total time period is fixed when T is assigned a particular value. Thus for smaller values of T, measurement data are collected over a smaller period of time (with m and n having specific values). For the example considered above, a choice of $T = 0.1$ msecond allows only $n(m+2)T=0.6$ mseconds for data collection, since this is well below the settling time of the system, a considerable amount of error in identified parameters is to be expected.

Although the sampling interval necessary for acceptable identification errors has been evaluated in this example through numerical results, for physical systems a-priori knowledge of its time-response / time-constant is not available.
Since the total number of samples to be taken is dictated by the mathematics, the sampling interval T may be chosen to fit in most of the measurement period over the settling time of the system.

For the second-order single-input single-output system, the number of measurement data is $n(m+2)T$ with a choice of sampling interval T there is a need to store only 6×6 measurement data in the array for matrix $V(T)$. Although this measurement data storage requirement is very modest, there is a need for a very large amount of space for matrix manipulation as indicated by Fig. 3-1. For the example considered here, a total of 190 kbytes of memory (IBM-4331,32-bit word-length data) was required for data storage and numerical computation. The total CPU time needed to complete the identification for this example was around 2.05 seconds. For small computers e.g., IBM-PC (IBM-5550 model), the storage requirement is not a problem, but the estimation error may increase due to shorter word length (typically 16-bit) unless the program was modified to perform calculations with 2-byte words. The processing time needed to compute the algorithm will increase with smaller computers due to architectural constraints on data and address buses.

Having established the general validity of the new identification algorithm, the following three examples are considered to find out if there is any relationship between the choice of sampling interval and the transient behavior of the true system, the three systems considered are underdamped, undamped and unstable, all having similar structure as the previous example.

Example 3 – 5 underdamped system: The true system is represented by

$$G(s) = \frac{1}{s^2 + s + 1}$$

which suggests that the dominant time constant of the system is around 2 seconds. For this system the total number of measurement data needed for each variable is n(m+2)T.

Allowing for 6 samples during the settling time period of the system, identification sampling interval (T) is 0.2 mseconds.

The identified parameters given by the algorithm for T = 0.2 mseconds (T_c/ 10,000) are

$$\hat{A} = \begin{bmatrix} -3.725290\,D-08 & 1.000000\,D+00 \\ -1.000401\,D+00 & -9.995987\,D-01 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} -4.011467\,D-04 \\ 1.000802\,D+00 \end{bmatrix}$$

$$\underline{\hat{x}}(t_o) = \begin{bmatrix} 0.985595\,D-01 \\ 1.802441\,D+00 \end{bmatrix}$$

For the true system, the parameters are :

$$A = \begin{bmatrix} 0.0 & 1.0 \\ -1.0 & -1.0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix}$$

$$\underline{x}(t_o) = \begin{bmatrix} 0.9 \\ 1.8 \end{bmatrix}$$

Thus the identification errors (2 − 49) and (3 − 71) are

$$S_A = 0.1180\,D-04$$

$$S_B = 0.1547\,D-05$$

$$S_O = 0.1120\,D-06$$

$$S = 0.7980\,D-05$$

The effects of varying T on S_A, S_B, S_O and S are shown in Fig. 3 − 4 , which supports the comments made at the end of Example 3 − 4 .

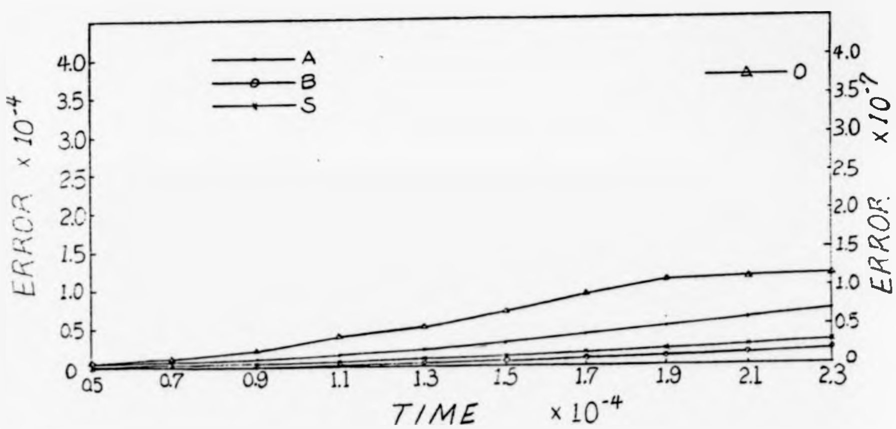

Fig. 3 — 4 The effects of varying T on identification errors (underdamped)

Example 3 − 6 undamped system : The true system is represented by

$$G(s) = \frac{s}{s^2 + 100}$$

which suggests that the oscillated period T_0 of the system is around 0.628 seconds. For this system the total number of measurement data needed for each variable is $n(m+2)T = 6T$, identification sampling interval (T) is 0.063 m seconds (T_0/ 10,000) .

The identified parameters given by the algorithm for T = 0.063 m seconds are

$$\hat{A} = \begin{bmatrix} -5.960464\,D-08 & 1.000000\,D+00 \\ -1.000004\,D+02 & 6.000936\,D-03 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} 9.997562\,D-01 \\ 6.371285\,D-03 \end{bmatrix}$$

$$\hat{x}(t_0) = \begin{bmatrix} -5.986589\,D-05 \\ 4.541631\,D-05 \end{bmatrix}$$

For the true system , the parameters are

$$A = \begin{bmatrix} 0.0 & 1.0 \\ -100.0 & 0.0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}$$

$$\underline{x}(t_0) = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$$

Thus the identification errors (2 − 49) and (3 − 71) are

$$S_A = 0.6382\,D-05$$

$$S_B = 0.1659\,D-04$$

$$S_O = 0.2823\,D-08$$

$$S = 0.9791\,D-05$$

The effects of varying T on S_A, S_B, S_O and S is shown in Fig. 3 − 5 , which supports the comments made at the end of Example 3 − 4 .

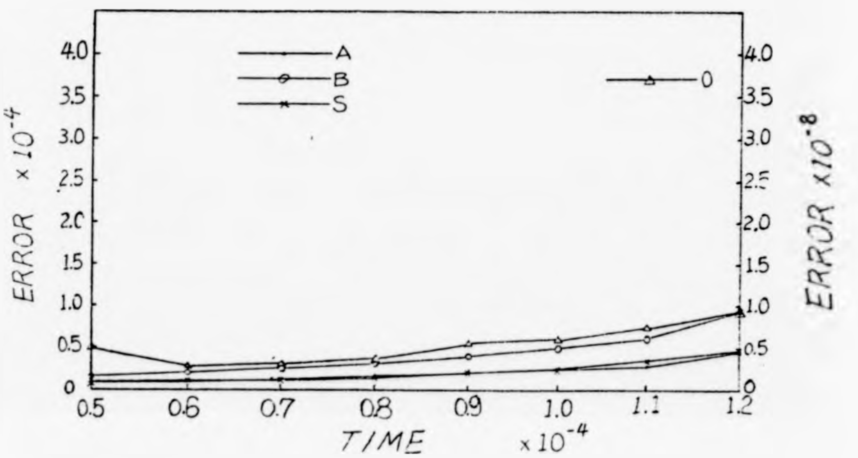

Fig. 3 − 5 The effects of varying T on identification errors (undamped)

Example 3 − 7 unstable system: The true system is represented by

$$G(s) = \frac{1.5}{s^2 + 4s - 17}$$

which suggests that the dominant time constant T_c of the system is around 0.33 seconds. For this system the total number of measurement data needed for each variable is $n(m+2)T = 6T$. Allowing for 6 samples of the system, identification sampling interval (T) is 0.15 mseconds.

The identified parameters given by the algorithm for $T = 0.15$ mseconds are

$$\hat{A} = \begin{bmatrix} 3.528595\text{D}-05 & 9.999981\text{D}-01 \\ 1.701221\text{D}+01 & -4.0036620\text{D}+00 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} 1.46484\text{D}-03 \\ 1.494705\text{D}+00 \end{bmatrix}$$

$$\hat{\underline{x}}(t_o) = \begin{bmatrix} 1.600333\text{D}+00 \\ -2.511990\text{D}+00 \end{bmatrix}$$

For the true system, the parameters are

$$A = \begin{bmatrix} 0.0 & 1.0 \\ 17.0 & -4.0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0 \\ 1.5 \end{bmatrix}$$

$$C = [\,1.0 \quad 0.0\,]$$

$$\underline{x}(t_0) = \begin{bmatrix} 1.6 \\ -2.5 \end{bmatrix}$$

Thus the identification errors (2 − 49) and (3 − 71) are

$$S_A = 0.3763\,D-05$$

$$S_B = 0.1504\,D-03$$

$$S_O = 0.1100\,D-04$$

$$S = 0.5507\,D-04$$

The effects of varying T on S_A, S_B, S_O and S is shown in Fig. 3 − 6 , which supports the comments made at the end of Example 3 − 4 .

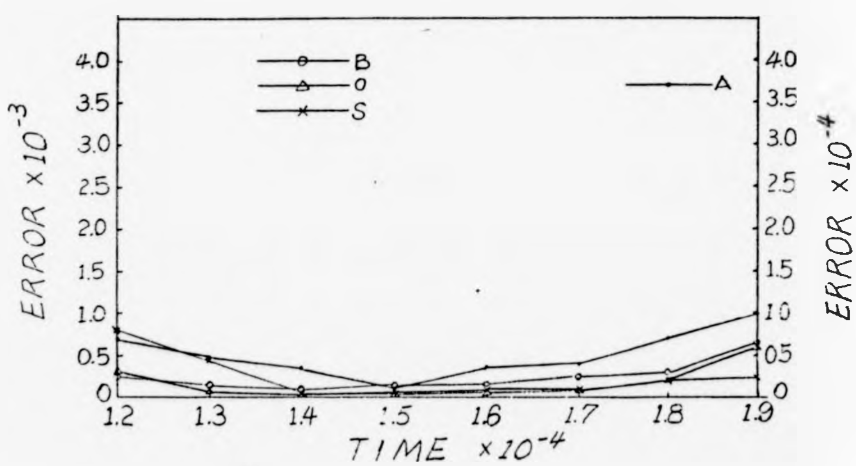

Fig. 3 − 6 The effects of varying T on identification errors (unstable)

3－5 Concluding remarks

We have considered a method to identify the parameters of a multiple-input multiple-output unknown order linear time invariant continuous system. The method, based on the multiple integration of dynamical equations, has the advantage of " filtering out " the effects of measurement noise on identification. The computational requirement of this method in very modest: one matrix inversion and one matrix multiplication.

An identification algorithm and computer program are derived to perform the numerical computation; observation of Fig. 3－3 , 3－4 , 3－5 and 3－6 related to the illustrative examples indicates that the agreement between the actual system parameters and estimated values is very close. In fact, the scalar error quantities S_A , S_B , S_0 and S are below 0.001 for all above examples over a wide range of sampling intervals. An important fact which comes out of the above figures in that the estimation errorrs are low at lower sampling intervals, this has considerable practical significance as the proposed algorithm reduces the need for a long period of time for data collection. In summary, it may be stated that the algorithm developed here is acceptable for application where a low-cost simple mechanism is needed for fast identification of system parameters from on-line data.

Perhaps the most important aspect of the proposed technique in that it can identify the parameters of overdamped, underdamped, undamped and unstable systems. Another advantage of this method is that it uses the system inputs and outputs and no special test signal. These make the method applicable for on-line estimation of parameter variations which may occur during the normal operation of the system under consideration, in addition to parameter identification. This method can also be used to obtain a low order model for a given high order system from experimental data, the theory and the illustrative examples are shown in Appendix 4.

Chapter—4 EXTENSION TO TIME-VARYING AND NON-LINEAR SYSTEMS

The previous two chapters were concerned with parameter estimation of linear time-invariant systems with known and unknown order. As illustrated through numerical examples, the proposed methods and the associated algorithms were found to be reasonably satisfactory with the main advantage being their relative simplicity. In view of these, it is worth exploring the possibilities of extending these results to non-linear and time-varying systems. The main advantage of such an extended theory is that the resulting algorithm could be applied to a very wide range of systems.

The derivations for such extensions have been based on the assumption that the state variables are accessible for measurement.

4 — 1 Time-varying systems

Consider the linear time-varying multiple-input multiple-output system S_t where

$$S_t: \quad \dot{\underline{x}}(t) = A(t)\underline{x}(t) + B(t)\underline{u}(t) \qquad (4-1)$$

$$\underline{y}(t) = C(t)\underline{x}(t) \qquad (4-2)$$

Where A(t), B(t) and C(t) are unknown time-varying matrices, and to be determined from experimental data.

In the derivations below, the matrices A(t), B(t) and C(t) are assumed to be Euler type, i.e., they are of the forms.

$$A(t) = \sum_{i=0}^{f} t^i A_i \ , \ B(t) = \sum_{i=0}^{l} t^i B_i \quad \text{and}$$

$$C(t) = \sum_{i=0}^{q} t^i C_i \qquad (4-3)$$

where $A_i \in R^{n\times n}$, $B_i \in R^{n\times m}$ and $C_i \in R^{p\times n}$ are unknown constant matrices to be determined.

Define :

$$\underline{x}_1(t) \triangleq \int_{t_0}^{t} \underline{x}(\tau)\,d\tau \qquad (4-4)$$

$$\underline{x}_{i+1}(t) \triangleq \int_{t_0}^{t} \underline{x}_i(\tau)\,d\tau \qquad (4-5)$$

$$\underline{u}_1(t) \triangleq \int_{t_0}^{t} \underline{u}(\tau)\, d\tau \tag{4-6}$$

$$\underline{u}_{i+1}(t) \triangleq \int_{t_0}^{t} \underline{u}_i(\tau)\, d\tau \tag{4-7}$$

$$\underline{w}(t) \triangleq \int_{t_0}^{t} \underline{y}(\tau)\, d\tau \tag{4-8}$$

$$\underline{G}_i(t) \triangleq \int_{t_0}^{t} t^i \underline{x}(\tau)\, d\tau \tag{4-9}$$

$$= \sum_{k=1}^{i+1} (-1)^{k+1} t^{i+1-k} \frac{i!}{(i+1-k)!} \underline{x}_k(t)$$

$$i = 0, 1, \cdots\cdots, f \tag{4-10}$$

$$\underline{K}_i(t) \triangleq \int_{t_0}^{t} t^i \underline{u}(\tau)\, d\tau$$

$$= \sum_{k=1}^{i+1} (-1)^{k+1} t^{i+1-k} \frac{i!}{(i+1-k)!} \underline{u}_k(t)$$

$$i = 0, 1, 2, \cdots\cdots, l \tag{4-11}$$

Assuming that all states are accessible for measurement, by integrating both sides of equation (4-1) and (4-2), we get :

$$\underline{x}(t) - \underline{x}(t_0) = \sum_{i=0}^{f} A_i \underline{G}_i(t) + \sum_{i=0}^{l} B_i \underline{K}_i(t) \tag{4-12}$$

$$\underline{w}(t) = \sum_{i=0}^{q} C_i \underline{G}_i(t) \tag{4-13}$$

Define :

$$H \triangleq \left[\begin{array}{cccc|cccc} A_0 & A_1 & \cdots\cdots\cdots\cdots & A_f & B_0 & B_1 & \cdots\cdots & B_l \\ \hline C_0 & C_1 & \cdots C_q \cdots 0 & \cdots 0 & 0 & 0 & \cdots\cdots & 0 \end{array}\right] \tag{4-14}$$

$$\underline{P}(t) \triangleq [\,\underline{G}_0(t) \quad \underline{G}_1(t)\cdots\cdots\underline{G}_f(t) \quad \underline{k}_0(t) \quad \underline{k}_1(t)\cdots\cdots\underline{k}_l(t)\,] \tag{4-15}$$

and

$$\underline{S}(t) = \left[\begin{array}{c} \underline{x}(t) - \underline{x}(t_0) \\ \hline \underline{w}(t) \end{array}\right] \tag{4-16}$$

where

$H \in R^{(n+p)\times(nf+ml)}$

$\underline{P}(t) \in R^{(nf+ml)\times 1}$

$\underline{S}(t) \in R^{(n+p)\times 1}$

In partioned matrix form, equations (4−12) and (4−13) can be written as :

$$\underline{S}(t) = H\underline{P}(t) \tag{4-17}$$

It is clear from equations (4−10) and (4−11) that to obtain $\underline{P}(t)$, we need to integrate the state vector $(f+1)$ successive times, and the input vector $(l+1)$ times successively.

Now if $\underline{x}(t)$, $\underline{x}_i(t)$, $\underline{u}_i(t)$ and $\underline{w}(t)$ are measured at $(nf+ml)$ successive equal intevals of time with a sampling period T, then equation (4−17) becomes :

$$S(T) = HP(T) \tag{4-18}$$

where

$$S(T) \triangleq [\,\underline{S}(T) \quad \underline{S}(2T)\cdots\cdots\underline{S}[(nf+ml)T]\,] \tag{4-19}$$

$$P(T) \triangleq [\, P(T) \;\; P(2T) \cdots\cdots P[(nf+ml)T]\,] \qquad (4-20)$$

$$S(T) \in R^{(n+p)\times(nf+ml)}$$

$$P(T) \in R^{(nf+ml)\times(nf+ml)}$$

If $P(T)$ is non-singular for some input $u(t)$ and a sampling period T, then the system is identifiable and the unknown matrices A_i, B_i and C_i can be computed as:

$$H = S(T)\; P(T)^{-1} \qquad (4-21)$$

If $P(T)$ is non-singular for $T = T_1$, and $T = T_2$ then resulted matrix H is unique.

Examples 4−1

Consider the following Euler-type linear time-varying system:

$$\dot{\underline{x}}(t) = \begin{bmatrix} 1+0.5t-2t^2 & 3+t \\ 2t+0.5t^2 & -2-t^2 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 1+3t \\ t \end{bmatrix} u(t)$$

$$y(t) = [\, 1+t \quad 2 \,]\, \underline{x}(t)$$

$$A_0 = \begin{bmatrix} 1 & 3 \\ 0 & -2 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0.5 & 1 \\ 2 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -2 & 0 \\ 0.5 & -1 \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

$$C_0 = [\, 1 \quad 2 \,], \quad C_1 = [\, 1 \quad 0 \,]$$

The identified parameters, i.e., $\hat{A}_i$, $\hat{B}_i$ and $\hat{C}_i$ obtained with different sampling periods are shown below.

(1) T = 1 second

$$\hat{A}_0 = \begin{bmatrix} 1.0015000\ E-00 & 2.9901201\ E+00 \\ 4.5959205\ E-04 & -2.0005907\ E+00 \end{bmatrix}$$

$$\hat{A}_1 = \begin{bmatrix} 4.9225341\ E-01 & 1.0020330\ E+00 \\ 1.5895514\ E+00 & 8.1303441\ E-04 \end{bmatrix}$$

$$\hat{A}_2 = \begin{bmatrix} -2.0000170\ E+00 & -2.9900000\ E-04 \\ 4.9948000\ E-01 & -1.0017000\ E+00 \end{bmatrix}$$

$$\hat{B}_0 = \begin{bmatrix} 9.9669802\ E-01 \\ -3.2776193\ E-04 \end{bmatrix}$$

$$\hat{B}_1 = \begin{bmatrix} 3.0006582\ E+00 \\ 1.0001850\ E+00 \end{bmatrix}$$

$$\hat{C}_0 = \begin{bmatrix} 9.9992232\ E-01 & 2.00000970\ E+00 \end{bmatrix}$$

$$\hat{C}_1 = \begin{bmatrix} 1.0000607\ E+00 & -1.0457914\ E-04 \end{bmatrix}$$

(2) T = 0.5 second

$$\hat{A}_0 = \begin{bmatrix} 9.9940977\ E-01 & 3.0028593\ E+00 \\ -4.4688238\ E-04 & -1.9977603\ E+00 \end{bmatrix}$$

$$\hat{A}_1 = \begin{bmatrix} 4.9034070\ E-01 & 0.9959712\ E-01 \\ 1.9996616\ E+00 & 2.4748177\ E-04 \end{bmatrix}$$

$$\hat{A}_2 = \begin{bmatrix} -2.0004732\ E+00 & 1.8665743\ E-04 \\ 4.9949517\ E-01 & -9.9994563\ E-01 \end{bmatrix}$$

$$\hat{B}_0 = \begin{bmatrix} 9.9995203\ E-01 \\ -5.8876104\ E-05 \end{bmatrix}$$

$$\hat{B}_1 = \begin{bmatrix} 3.0007522\ E+00 \\ 1.0007728\ E+00 \end{bmatrix}$$

$$\hat{C}_0 = \begin{bmatrix} 1.0000258\ E+00 & 1.9997565\ E+00 \end{bmatrix}$$

$$\hat{C}_1 = \begin{bmatrix} 1.0000448\ E+00 & 7.1793911\ E-05 \end{bmatrix}$$

4 — 2 Non-linear systems

Consider the non-linear system represented by the differential equation :

$$\sum_{i=0}^{n} a_i y(t)^{(i)} + N(y) = \sum_{j=0}^{m} b_j u^{(j)}(t) \qquad (4-22)$$

where $y(t)$ and $u(t)$ are the system output and input respectively.

a_i , b_j are the system parameters to be determined.

$N(y)$ is the non-linear component of the system.

If the form of the non-linearity in the non-linear differential equation (4 — 22) is known, and if equation(4—22) can be integrated in terms of the system output or its successive integrals, then with some modifications, the earlier method can be applied to the unknown parameters of such systems. Here the identification algorithm is not general as in the case of linear systems, different algorithm are need to cope with different forms of the non-linear components; The usefulness of the general method is best demonstrated by the following example.

Example 4—2

Consider the non-linear system governed by Duffing's equation :

$$\ddot{y}(t)+a_1 y(t)+a_2 y^3(t)=b_1 u(t) \qquad (4-23)$$

where a_1, a_2 and b_1 are unknown constants to be determined.

This system was considered by T.C. Hsia and V. Vimalvanich [10], the identification method they used is based on the learning model concept, where a time-varying model is required. The true parameter values used were $a_1=3$, $a_2=0.5$ and $b_1=3$.

Define :

$$z_1(t) \triangleq \int_{t_0}^{t}\int_{t_0}^{t} y(t)\,dt$$

$$z_2(t) \triangleq \int_{t_0}^{t}\int_{t_0}^{t} y^3(t)\,dt$$

and

$$V(t) \triangleq \int_{t_0}^{t}\int_{t_0}^{t} u(t)\,dt$$

Then if we integrate both sides of equation (4—23) 2-times we get :

$$y(t) - y(t_0) - (t - t_0)\dot{y}(t_0) + a_1 z_1(t) + a_2 z_2(t)$$
$$= b_1 V(t) \tag{4-24}$$

Define :

$$H \triangleq [\,-a_1 \quad -a_2 \quad b_1 \quad \dot{y}(t_0)\,]$$

and

$$\underline{P}(t) \triangleq [\, z_1(t) \quad z_2(t) \quad V(t) \quad (t - t_0)\,]^T$$
$$\ell(t) \triangleq y(t) - y(t_0)$$

where

$$H^T \in R^4 \ , \ \underline{P}(t) \in R^4$$

Then equation (4—24) can be written as :

$$\ell(t) = H\underline{P}(t) \tag{4-25}$$

Now if $\ell(t)$, $\underline{P}(t)$ are measured at 4 - successive sampling periods, then

$$\underline{\ell}(T) = HP(T) \tag{4-26}$$

where

$$\underline{\ell}(T) = [\, \ell(T) \quad \ell(2T) \quad \ell(3T) \quad \ell(4T)\,]$$
$$P(T) = [\, \underline{P}(T) \quad \underline{P}(2T) \quad \underline{P}(3T) \quad \underline{P}(4T)\,]$$
$$\underline{\ell}^T(T) \in R^4$$
$$P(T) \in R^{4\times 4}$$

Now if P(T) is non-singular, then the unknown parameters and the initial condition can be determined as :

$$H = \underline{x}(T)P(T)^{-1}$$
$$= [\,-a_1 \quad -a_2 \quad b_1 \quad \dot{y}(t_0)\,]$$

4 — 3 Concluding remarks

The methods of previous two chapters have been extended to non-linear and time-varying systems. In the time-varying case, the matrices A(t), B(t) and C(t) are assumed to be in certain special forms (the Euler type). If the time-varying elements of the system are known to be slowly varying, i.e., the parameters remain constant over the time interval $[t_0, t_f]$, then the system can be considered as time invariant and both methods of Chapter 2 and 3 can be applied at successive intervals of time by resetting the integrators at the end of each interval. In non-linear case the system is assumed to be in the form of Duffing's equation. The general philosophy of integrating the dynamical equations is applied to this case. With some modification the method of Chapter 3 may be used for the identification of time-varying and non-linear systems.

Chapter 5 DISCRETE SISO SYSTEMS

This chapter considers the problem of identification of discrete single-input single-output (SISO) systems, primarily to complement the previous development for continuous systems. The problem is considered in two parts, in first part the order of the system is determined, while in the second part, the system parameters are to be identified. Numerical examples are presented in each case to illustrate the application of the algorithms developed. The main objective here is to develop a mathematical framework such that both discrete and continuous linear systems may be identified by the same parameter estimation package.

5 — 1 Order determination and flow chart

Consider the nth order single-input single-output linear time-invariant discrete-time system S_1

$$S_1 : \underline{x}(k+1) = \Phi\,\underline{x}(k) + F\,u(k) \tag{5-1}$$

$$y(k) = G\,\underline{x}(k) + D\,u(k) \tag{5-2}$$

where

k is the sampling instant t_k, and is an integer on $[0, N]$, $N < \infty$

$\underline{x}(k) \in R^{n \times 1}$ is the state vector

$y(k)$ is output

$u(k)$ is input

$\Phi \in R^{n \times n}$ is the system matrix

$F \in R^{n \times 1}$ is the input column

$G \in R^{1 \times n}$ is the output row

D is scaler, n is system order

Let

$$E(N+1) \triangleq \{ u(k) \in R,\ y(k) \in R\ ;\ k = 0, 1, \cdots\cdots, N \}$$

be a sequence of $N+1$ input-output measurements from the system S_1

PROBLEM: Based on the finite input-output measurement $E(N+1)$, find the order n.

By consecutive substitutions of (5—1) into (5—2), we have:

$\underline{j=1}$
$$\begin{aligned} y(k+1) &= G\underline{x}(k+1) + Du(k+1) \\ &= G[\Phi\underline{x}(k) + F u(k)] + Du(k+1) \\ &= G\Phi\underline{x}(k) + GF u(k) + Du(k+1) \end{aligned}$$

$\underline{j=2}$
$$\begin{aligned} y(k+2) &= G\underline{x}(k+2) + Du(k+2) \\ &= G[\Phi\underline{x}(k+1) + Fu(k+1)] + Du(k+2) \\ &= G\Phi^2\underline{x}(k) + G\Phi F u(k) + GF u(k+1) + Du(k+2) \end{aligned}$$

$\vdots$

$\underline{j=n-1:}$ $\quad y(k+n-1)=G\Phi^{n-1}\underline{x}(k)+G\Phi^{n-2}F\ u(k)+\cdots\cdots+Du(k+n-1)$

Finally, we obtain

$$y(k+j)=G\Phi^{j}\underline{x}(k)+\sum_{i=0}^{j-1}G\Phi^{j-i-1}F\ \ u(k+i)+Du(k+j) \tag{5-3}$$

Letting $j=0,1,2,\cdots\cdots,n-1$ in (5−3) and concatenating the n equations, we have

$$\begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ \vdots \\ y(k+n-1) \end{bmatrix} = \begin{bmatrix} G \\ G\Phi \\ \vdots \\ \vdots \\ G\Phi^{n-1} \end{bmatrix} \underline{x}(k) + \begin{bmatrix} D & 0 & & & 0 \\ GF & D & & & \\ G\Phi F & GF & D & & \\ \vdots & \vdots & \ddots & \ddots & \\ G\Phi^{n-2}F & G\Phi^{n-3}F & G\Phi^{n-4}F\cdots & GF & D \end{bmatrix} \cdot \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ \vdots \\ u(k+n-1) \end{bmatrix} \tag{5-4}$$

where

$[\,y(k)\quad y(k+1)\cdots\cdots y(k+n-1)\,]^{T}\in R^{n\times 1}$

$[\,G\quad G\Phi\cdots\cdots\cdots\cdots\cdots\cdots G\Phi^{n-1}\,]^{T}\in R^{n\times n}$

$\underline{x}(k)\in R^{n\times 1}$

$[\,u(k)\quad u(k+1)\cdots\cdots u(k+n-1)\,]^{T}\in R^{n\times 1}$

and

$$\begin{bmatrix} D & 0 & & \\ GF & D & & 0 \\ G\Phi F & GF & \ddots & \\ \vdots & \vdots & \ddots & \ddots \\ G\Phi^{n-2}F & G\Phi^{n-3}F & \cdots GF & D \end{bmatrix} \in R^{n\times n}$$

If system S_1 is completely observable and system order is n, then

$$\text{rank}\begin{bmatrix} G \\ G\Phi \\ \vdots \\ G\Phi^{n-1} \end{bmatrix} = n \qquad (5\text{—}5)$$

and there exist a set of n real numbers, $\{ a_i : i=0,1,\cdots,n-1 \}$ such that for all $k > n$

$$\sum_{i=0}^{n} a_i G\Phi^{k-n-i} = 0 \ , \ a_n = 1 \qquad (5\text{—}6)$$

Thus, there exist two nonsingular $n\times n$ matrices M_1 and M_2 such that

$$M_1 \begin{bmatrix} G \\ G\Phi \\ \vdots \\ G\Phi^{n-1} \end{bmatrix} M_2 = I_n \qquad (5\text{—}7)$$

where

$$M_1 \in R^{n\times n} \quad , \quad M_2 \in R^{n\times n}$$

define:

$$\underline{E}(k)=\begin{bmatrix} E_1(k) \\ E_2(k) \\ \vdots \\ E_n(k) \end{bmatrix} = M_2^{-1}\,\underline{x}(k) \tag{5-8}$$

where

$$\underline{E}(k) \in R^{n\times 1}$$

pre-multiplying(5-4) by M_1 , We obtain the following equation:

$$M_1\begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+n-1) \end{bmatrix} = M_1\begin{bmatrix} G \\ G\Phi \\ \vdots \\ G\Phi^{n-1} \end{bmatrix} M_2\,\underline{E}(k) + M_1 \cdot \begin{bmatrix} D & 0 & & & \\ GF & D & 0 & & \\ G\Phi F & GF & D & & \\ \vdots & \vdots & \ddots & \ddots & \\ G\Phi^{n-2}F & G\Phi^{n-3}F & \cdots & GF & D \end{bmatrix} \cdot \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+n-1) \end{bmatrix} \tag{5-9}$$

Let

$$M_1 = \begin{bmatrix} & & \\ & In & \\ & & \end{bmatrix} \tag{5-10}$$

then

$$M_1^{-1} = \begin{bmatrix} & & \\ & In & \\ & & \end{bmatrix} \tag{5-11}$$

Then by substituting (5—7), (5—8) and (5—11) into (5—9), we have :

$$\begin{bmatrix} E_1(k) \\ E_2(k) \\ \vdots \\ E_n(k) \end{bmatrix} = \begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+n-1) \end{bmatrix} - \begin{bmatrix} D & & & & \\ GF & D & & 0 & \\ G\Phi F & GF & \ddots & & \\ \vdots & \vdots & \ddots & \ddots & \\ G\Phi^{n-2}F & G\Phi^{n-3}F & \cdots & GF & D \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+n-1) \end{bmatrix} \tag{5-12}$$

From (5—6) ~ (5—8) and (5—10) 、(5—11), the first term of (5—3) for $j = n$, $G\Phi^n \underline{x}(k)$, can expressed as :

$$G\Phi^n \underline{x}(k) = \left(\sum_{i=0}^{n-1} - a_i G\Phi^i \right) M_2 \cdot M_2^{-1} \underline{x}(k)$$

$$= [\, -a_0 \quad -a_1 \, \cdots\cdots \, -a_{n-1} \,] \begin{bmatrix} G \\ G\Phi \\ \vdots \\ G\Phi^{n-1} \end{bmatrix} M_2 \underline{E}(k)$$

$$
= \left[-\alpha_0 \quad -\alpha_1 \cdots\cdots -\alpha_{n-1} \right] M_1^{-1} M_1 \begin{bmatrix} G \\ G\Phi \\ \vdots \\ G\Phi^{n-1} \end{bmatrix} M_2 \underline{E}(k)
$$

$$
= \left[-\alpha_0 \quad -\alpha_1 \cdots\cdots -\alpha_{n-1} \right] \begin{bmatrix} E_1(k) \\ E_2(k) \\ \vdots \\ E_n(k) \end{bmatrix} \qquad (5-13)
$$

Thus, by letting $j = n$ in equation (5 − 3) and by using equations (5 − 12) and (5 − 13), the system S can be expressed as following :

$$
y(k+n) = \left[-\alpha_0 \quad -\alpha_1 \cdots\cdots -\alpha_{n-1} \right] \begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+n-1) \end{bmatrix} - \left[-\alpha_0 \quad -\alpha_1 \cdots -\alpha_{n-1} \right] \cdot
$$

$$
\begin{bmatrix} D & & & 0 \\ GF & D & & \\ \vdots & GF & \ddots & \\ G\Phi^{n-2}F & \vdots & \ddots & \\ & G\Phi^{n-3}F \cdots & GF & D \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+n-1) \end{bmatrix} + \sum_{i=0}^{n-1} G\Phi^{n-i-1} F u(k+i)
$$

$$
+ Du(k+n)
$$

$$
= -\sum_{i=0}^{n-1} \alpha_i y(k+i) + \sum_{i=0}^{n} \beta_i u(k+i) \qquad (5-14)
$$

where

a_i , $i = 0, 1, 2, \cdots\cdots, n-1$ are real number defined by equation (5−6)

$$\beta_i \triangleq \begin{cases} a_i D + \sum_{j=i+1}^{n} a_j G \Phi^{j-1} F & \text{for } 0 \leq i \leq n-1 \\ a_n D & \text{for } i = n \end{cases}$$

Define

$$\underline{R}(k) \triangleq [\ y(0) \quad y(1) \cdots\cdots y(k) \quad u(0) \quad u(1) \cdots\cdots u(k)\ ]$$

where

$$\underline{R}(k) \in R^{1 \times (2k+2)}$$

$$A \triangleq \begin{bmatrix} \underline{R}(k) \\ \underline{R}(k+1) \\ \vdots \\ \underline{R}(N) \end{bmatrix} \triangleq \left[\begin{array}{ccc|c|ccc} y(0) & \cdots\cdots & y(k-1) & y(k) & u(0) & \cdots\cdots & u(k) \\ y(1) & \cdots\cdots & y(k) & y(k+1) & u(1) & \cdots & u(k+1) \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ y(N-k) & \cdots & y(N-1) & y(N) & u(N-k) & \cdots & u(N) \end{array} \right] \tag{5−15}$$

where

$$A \in R^{(N-k+1) \times (2k+2)}$$

and

$$B \triangleq \left[\begin{array}{ccc|ccc} y(0) & \cdots\cdots & y(k-1) & u(0) & \cdots\cdots & u(k) \\ y(1) & \cdots\cdots & y(k) & u(1) & \cdots\cdots & u(k+1) \\ \vdots & & \vdots & \vdots & & \vdots \\ y(N-k) & \cdots & y(N-1) & u(N-k) & \cdots & u(N) \end{array} \right] \tag{5−16}$$

where

$$B \in R^{(N-k+1) \times (2k+1)}$$

Let

$E(N+1) \triangleq \{ u(k) \in R,\ y(k) \in R\ ;\ k=0,1,\cdots\cdots,N \}$ be a sequence of input-output measurements obtained from the system S.

Consider the following equation

$$
\begin{aligned}
DIF &\triangleq \text{rank}(A) - \text{rank}(B) \\
&= \text{rank}\left[\begin{array}{ccc|c|ccc}
y(0) & \cdots\cdots & y(k-1) & y(k) & u(0) & \cdots\cdots & u(k) \\
y(1) & \cdots\cdots & y(k) & y(k+1) & u(1) & \cdots\cdots & u(k+1) \\
\vdots & & \vdots & \vdots & \vdots & & \vdots \\
y(N-k) & \cdots & y(N-1) & y(N) & u(N-k) & \cdots & u(N)
\end{array}\right] \\
&\quad - \text{rank}\left[\begin{array}{ccc|ccc}
y(0) & \cdots\cdots & y(k-1) & u(0) & \cdots\cdots & u(k) \\
y(1) & \cdots\cdots & y(k) & u(1) & \cdots\cdots & u(k+1) \\
\vdots & & \vdots & \vdots & & \vdots \\
y(N-k) & \cdots & y(N-1) & u(N-k) & \cdots & u(N)
\end{array}\right] \qquad (5-17)
\end{aligned}
$$

Note that the only difference between the two matrices in (5-17) is that A contains one more column than B; this column is

$$\underline{y}(k;N) \triangleq [\, y(k)\ \ y(k+1)\ \cdots\cdots\ y(N)\,]^T \qquad (5-18)$$

where

$$\underline{y}(k;N) \in R^{(N-k+1)\times 1}$$

Thus, DIF can only assume the value of either 0 or 1 for all non-negative integers k. Based on the values of DIF, we can make the following conclusions :

(1) If $DIF = 0$ for some non-negative integer k, then the column vector $\underline{y}(k ; N)$ is linearly dependent on the columns of matrix B. Thus, a linear, time-invariant, discrete-time system of order n of the form:

$$y(k+n) = -\sum_{i=0}^{n-1} a_i y(k+i) + \sum_{i=0}^{n} \beta_i u(k+i) \qquad ((5-14))$$

satisfies the input-output time series relationship

$$E(N-n+1) \triangleq \{ u(k), y(k); k = 0, 1, 2, \cdots\cdots N-n \}$$

(2) If $DIF = 1$, then the column vector $\underline{y}(k ; N)$ is linearly independent of all the columns of matrix B (5-16). Thus, no linear, time-invariant, discrete-time systems of order n of the form (5-14) can be satisfied by the partial time-series $E(N-n+1)$ above.

Here we conclude the procedure for identified algorithm step by step:

<u>Step 1</u> Start with $k = 0$

<u>Step 2</u> Construct the matrices A and B from the available data $E(N+1)$ according to equations (5-15) and (5-16).

<u>Step 3</u> Compute DIF according to equation (5-17).

<u>Step 4</u> (1) If $DIF = 1$, then $n > k$ and increase k by one and go to step 2.

(2) If $DIF = 0$, then $n = k$

These steps can used to develop a numerical algorithm for the identification of single-input single-output systems. The flow-chart of this algorithm is shown in Fig. 5–1 and a computer listing of the program is given in Appendix 3 (Program C). A numerical example to demonstrate the validity of this identification software is given in the following section.

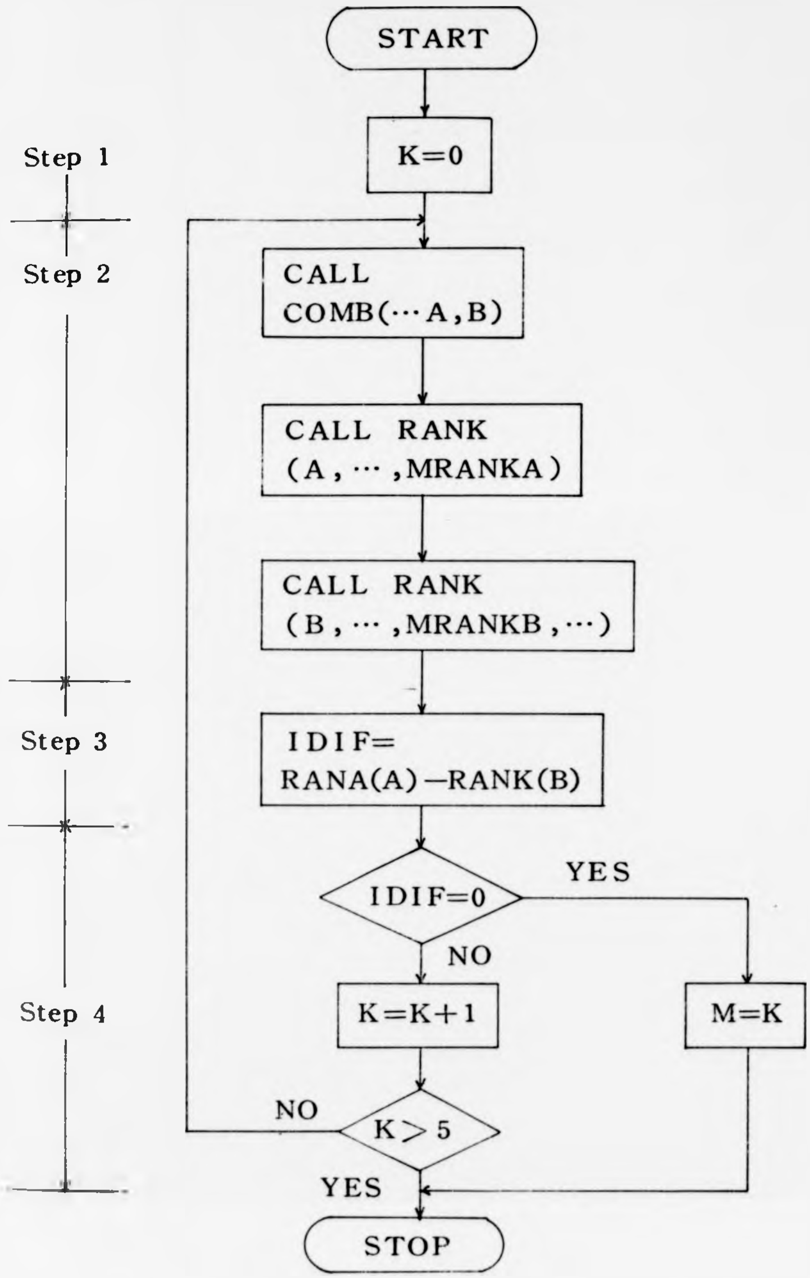

Fig. 5—1 The flow chart of computer program to determine the system order.

5 — 2 Numerical examples of order determination

To demonstrate the use of the method to determine the order of discrete-time linear systems, we consider the following examples. For the numerical computation the computer program (program C) in Appendix 3 was used.

(a) Example 1

In this example, we consider a 2nd order difference equation

$$y(k+2) = -y(k) + y(k+1) - u(k) + u(k+1) + 0\,u(k+2)$$

where

$$u(kT) = \begin{cases} 1.0 + 2k \sin(60kT) & k > 0 \\ 1 & k = 0 \\ 0 & k < 0 \end{cases}$$

$n = 2$, sampling length $N = 19$

The identified results are shown below.

$K = 0$

RANK $A = 2$, RANK $B = 1$, DIF = 1

$K = 1$

RANK $A = 4$, RANK $B = 3$, DIF = 1

$K = 2$

RANK $A = 5$, RANK $B = 5$, DIF = 0

The system order = 2

(b) Example 2

In this example, we consider a 3rd order difference equation

$$y(k+3) = -2y(k) - 1.5y(k+1) - y(k+2) + 2u(k) + u(k+1) + 0.5u(k+2)$$

where

$n = 3$

$$u(kT) = \begin{cases} 1 + k\sin(60kT) & k > 0 \\ 1 & k = 0 \\ 0 & k < 0 \end{cases}$$

The sampling length $N = 19$

The identified results are shown below.

$k = 0$

RANK $A = 2$, RANK $B = 1$, DIF = 1

k = 1

RANK A = 5 , RANK B = 4 , DIF = 1

k = 2

RANK A = 6 , RANK B = 5 , DIF = 1

k = 3

RANK A = 7 , RANK B = 7 , DIF = 0

The system order = 3

5 − 3 Parameters determination and flow-chart

Consider the system S_1 as described by (5—12).

$$y(k+n) = -\sum_{i=0}^{n-1} a_i y(k+i) + \sum_{i=0}^{n} \beta_i u(k+i) \qquad ((5-12))$$

where n is the order of S_1 determined by applying the method that has been developed in section 5−1, the vector $\underline{y}(k;N)$ as defined by (5−18), i.e.,

$$\underline{y}(k;N) = [\, y(k) \;\; y(k+1) \cdots\cdots y(N) \,]^T \qquad ((5-18))$$

is a linear combination of the columns of the matrix B [equation (5—16)].

Let $E(N+1) \triangleq \{ u(k) \in R \;\; y(k) \in R \,;\, K = 0, 1, \cdots\cdots, N \}$ be an available sequence of input-output measurements, and

$$ALPHA \triangleq [\, -a_0, -a_1, \cdots\cdots, -a_{n-1}, \beta_0, \beta_1, \cdots\cdots, \beta_n \,]^T$$

be a $(2k+1) \times 1$ parameter vector comprising the coefficients of this linear combination. Then we can write

$$\underline{y}(k;N) = B \cdot ALPHA =$$

$$\left[\begin{array}{cccc|ccc} y(0) & y(1) & \cdots\cdots\cdots & y(k-1) & u(0) & \cdots\cdots & u(k) \\ y(1) & y(2) & \cdots\cdots\cdots & y(k) & u(1) & \cdots\cdots & u(k+1) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ y(N-k) & y(N-k+1) & \cdots\cdots & y(N-1) & u(N-k) & \cdots\cdots & u(N) \end{array}\right] \cdot$$

$$[\, -a_0, -a_1, \cdots\cdots, -a_{n-1}, \beta_0, \beta_1, \cdots\cdots, \beta_n \,]^T \qquad (5-19)$$

Define:

$$H \triangleq \frac{1}{N-k+1} A^{T} A$$

$$
= \frac{1}{N-k+1}
\left[\begin{array}{cccc|cccc}
\sum_{n=0}^{N-k} y^{2}(n) & \sum_{n=0}^{N-k} y(n)y(n+1) & \cdots & \sum_{n=0}^{N-k} y(n)y(k+n) & \sum_{n=0}^{N-k} y(n)u(n) & \sum_{n=0}^{N-k} y(n)u(n+1) & \cdots & \sum_{n=0}^{N-k} y(n)u(k+n) \\
 & \sum_{n=1}^{N-k+1} y^{2}(n) & \cdots & \sum_{n=1}^{N-k+1} y(n)y(k+n-1) & \sum_{n=1}^{N-k+1} y(n)u(n-1) & \sum_{n=1}^{N-k+1} y(n)u(n) & \cdots & \sum_{n=1}^{N-k+1} y(n)u(k+n-1) \\
 & & \ddots & \vdots & \vdots & \vdots & & \vdots \\
 & & & \sum_{n=k}^{N} y^{2}(n) & \sum_{n=k}^{N} y(n)u(n-k) & \vdots & & \sum_{n=k}^{N} y(n)u(n) \\
\hline
 & & & & \sum_{n=0}^{N-k} u^{2}(n) \cdots & \sum_{n=0}^{N-k} u(n)u(n+1) & \cdots & \sum_{n=0}^{N-k} u(n)u(k+n) \\
 & & 0 & & & \sum_{n=1}^{N-k+1} u^{2}(n) & & \sum_{n=1}^{N-k+1} u(n)u(k+n-1) \\
 & & & & & & \ddots & \vdots \\
 & & & & & & & \sum_{n=k}^{N} u^{2}(n)
\end{array}\right]
$$

(5−20)

where

$H \in R^{(2k+2)\times(2k+2)}$

and

$$P \triangleq \frac{1}{N-k+1} B^T B$$

$$= \frac{1}{N-k+1}\left[\begin{array}{cccc} \sum_{n=0}^{N-k} y^2(n) & \sum_{n=0}^{N-k} y(n)y(n+1) & \cdots & \sum_{n=0}^{N-k} y(n)y(k+n-1) \\ & \sum_{n=1}^{N-k+1} y^2(n) & \cdots & \sum_{n=1}^{N-k+1} y(n)y(k+n-2) \\ & & \ddots & \vdots \\ & & & \sum_{n=k-1}^{N-1} y^2(n) \\ \hline & & 0 & \end{array}\right.$$

$$\left.\begin{array}{cccc} \sum_{n=0}^{N-k} y(n)u(n) & \sum_{n=0}^{N-k} y(n)u(n+1) & \cdots & \sum_{n=0}^{N-k} y(n)u(k+n) \\ \sum_{n=1}^{N-k+1} y(n)u(n-1) & \sum_{n=1}^{N-k+1} y(n)u(n) & \cdots & \sum_{n=1}^{N-k+1} y(n)u(k+n-1) \\ \vdots & \vdots & & \vdots \\ \hline \sum_{n=0}^{N-k} u^2(n) & \sum_{n=0}^{N-k} u(n)u(n+1) & \cdots & \sum_{n=0}^{N-k} u(n)u(k+n) \\ & \sum_{n=1}^{N-k+1} u^2(n) & \cdots & \sum_{n=1}^{N-k+1} u(n)u(k+n-1) \\ & & \ddots & \vdots \\ & & & \sum_{n=k}^{N} u^2(n) \end{array}\right] \quad (5-21)$$

where

$$P \in R^{(2k+1)\times(2k+1)}$$

Multiply $\frac{1}{N-k+1}B^T$ on both sides of equation (5−19):

$$P\cdot ALPHA=\frac{1}{N-k+1}B^T\underline{y}(k;N)$$

$$=\left[\begin{array}{c|c|c} I_k & 0 & 0 \\ \hline \multicolumn{2}{c|}{0} & I_{k+1}\end{array}\right]\frac{1}{N-k+1}A^TA\begin{array}{c}\left[\begin{array}{c}0\\0\\\vdots\\0\\1\\0\\0\\\vdots\\0\end{array}\right]\end{array}\begin{array}{l}\\\\\\\\(k+1)\text{th row}\\\\\\\\\\\end{array}$$

$$=\left[\begin{array}{c|c|c} I_k & 0 & 0 \\ \hline \multicolumn{2}{c|}{0} & I_{k+1}\end{array}\right]\cdot H\cdot\left[\begin{array}{c}0\\0\\\vdots\\0\\1\\0\\\vdots\\0\end{array}\right] \tag{5−22}$$

where

$$\left[\begin{array}{c|c|c} I_k & 0 & 0 \\ \hline \multicolumn{2}{c|}{0} & I_{k+1}\end{array}\right]\in R^{(2k+1)\times(2k+2)},\quad [0\,0\cdots100\cdots0]^T\in R^{(2k+2)\times1}$$

If the matrix P is non-singular, then

$$ALPHA=P^{-1}\cdot\left[\begin{array}{c|c|c} I_k & 0 & 0 \\ \hline \multicolumn{2}{c|}{0} & I_{k+1}\end{array}\right]\cdot H\cdot\left[\begin{array}{c}0\\0\\\vdots\\0\\1\\0\\0\\\vdots\\0\end{array}\right] \tag{5−23}$$

The above derivations lead to the following procedure for estimation of system parameters.

Step 1 Consider the nth order single-input single-output linear time-invariant discrete-time system S_1

$$S_1 : y(k+n) = -\sum_{i=0}^{n-1} \alpha_i y(k+i) + \sum_{i=0}^{n} \beta_i u(k+i) \quad ((5\text{--}14))$$

Step 2 Construct matrix A from equation (5–15)

Step 3 Compute matrix A^T.

Step 4 Construct matrix B from equation (5–16) and Compute B^T.

Step 5 Compute matrix H from equation (5–20).

Step 6 Compute matrix P from equation (5–21).

Step 7 Compute matrix J.

where

$$J \triangleq \left[\begin{array}{c|c|c} I_M & 0 & 0 \\ \hline \multicolumn{2}{c|}{0} & I_{M+1} \end{array}\right]$$

M is the estimated system order

Step 8 Compute the vector E.

where

$$E \triangleq \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \\ \\ \\ (M+1)\text{ th row} \\ \\ \\ \\ \end{array}$$

Step 9 Compute the system parameter ALPHA from equation (5—23) .

(Program C) for the above algorithm is shown in Fig. 5—2 and the listing of the program (program C) is shown in Appendix 3 .

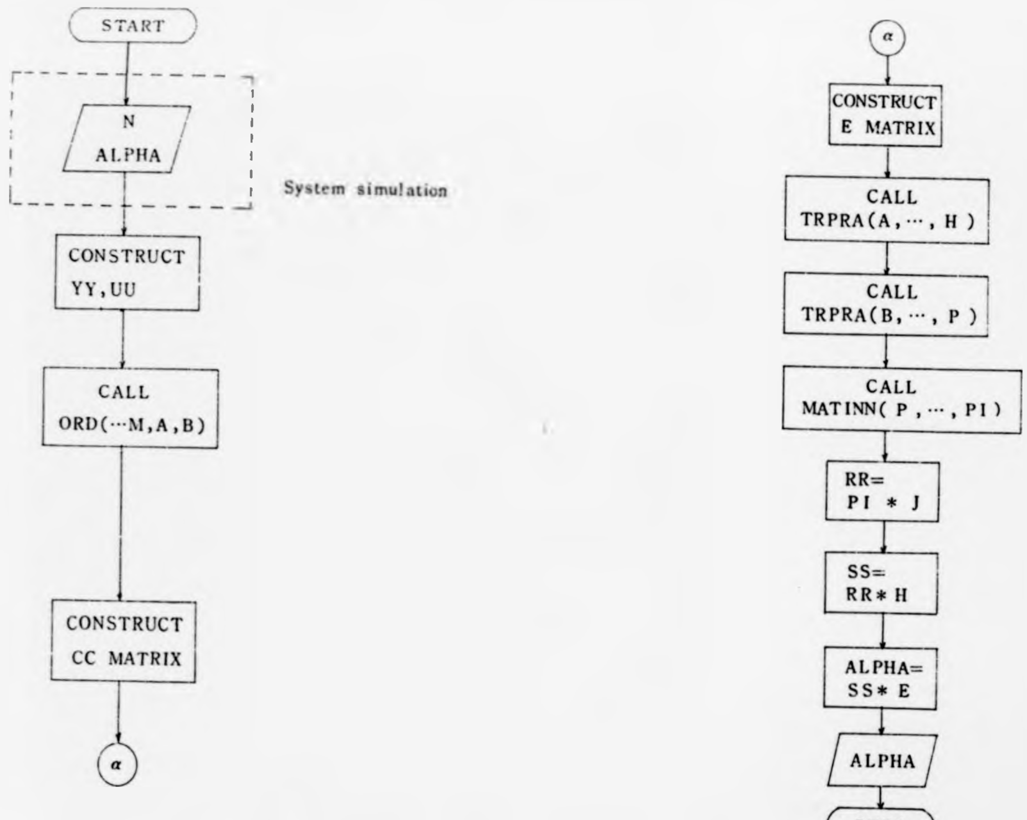

Fig. 5−2 The flow-chart of computer program to identify the parameter of discrete-time system

The variables used in the Fig. 5−2 are defined as :

N is the number of samples,

UU is the vector of input sequence

$$UU = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N) \end{bmatrix}$$

YY is the vector of output sequence

$$YY = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N) \end{bmatrix}$$

ALPHA is computed parameter

ALPHA = [AK(1) …… AK(n) BK(1) …… BK(n+1)]

M is determined system order equal to n

$$J \triangleq \left[\begin{array}{cc|c} I_M & 0 & 0 \\ \hline \multicolumn{2}{c|}{0} & I_{M+1} \end{array}\right]$$

$$E \triangleq \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} (M+1)\text{th row}$$

5 — 4 Numerical examples of parameters determination

To illustrate the use of the above method to estimate the parameters of discrete-time linear systems, we consider the following examples. For the numerical computation, the computer program (Program C) in Appendix 3 was used.

(a) Example 1

Consider the 2nd order difference equation

$$y(k+2) = -y(k) + y(k+1) - u(k) + u(k+1) + 0\,u(k+2)$$

where

$$u(k) = \begin{cases} 1 + 2k \sin(60kT) & k > 0 \\ 1 & k = 0 \\ 0 & k < 0 \end{cases}$$

$T = 1$ second

the original parameter ALPHA is

$$\text{ALPHA} = [\,-1 \quad 1 \quad -1 \quad 1 \quad 0\,]^T$$

and

System order $n = 2$.

Assuming that the system order n is determined by applying the method developed in section 5—1 ; five different values of N were chosen to identify the ALPHA .
The corresponding results are

(1) N = 19

ALPHA = 〔 −1.000000 E+00　1.000000 E+00　−1.000000 E+00　1.000000 E+00　4.618528 E−14 〕^{T}

(2) N = 29

ALPHA = 〔 −9.999999 E−01　9.999999 E−01　−9.999999 E−01　9.999999 E−01　−3.197442 E−14 〕^{T}

(3) N = 39

ALPHA = 〔 −9.999999 E−01　9.999999 E−01　−9.999999 E−01　9.999999 E−01　−4.618528 E−14 〕^{T}

(4) N = 49

ALPHA = 〔 −1.000000 E+00　1.000000 E+00　−1.000000 E+00　1.000000 E+00　5.115908 E−13 〕^{T}

(5) N = 59

ALPHA = 〔 −1.000000 E+00　1.000000 E+00　−1.000000 E+00　1.000000 E+00　2.842171 E−13 〕^{T}

From these results, the estimation errors S_A and S_B are acceptable for most analysis/design studies using linear theory.

(b) Example 2

Consider the 4th order linear discrete-time system

$$y(k+4) = -6.6\times10^{-3}y(k) + 0.4888\,y(k+1) - 1.7512y(k+2) + 2.2605\,y(k+3) + 6.6\times10^{-3}\,u(k) - 0.1245\,u(k+1) + 2.89\times10^{-2}\,u(k+2) + 9.75\times10^{-2}\,u(k+3)$$

The results for five different samples are shown given below.

(1) N = 19

THE ALPHA VECTOR =

```
-6.599996E-03  4.887999E-01  -1.751199E+00  2.260499E+00  6.599996E-03  -1.244999E-01
 2.890000E-02  9.749997E-02   3.505876E-15
```

(2) N = 29

THE ALPHA VECTOR=

```
-6.599996E-03  4.887999E-01  -1.751199E+00  2.260499E+00  6.599996E-03  -1.244999E-01
 2.890000E-02  9.749997E-02   1.690315E-14
```

(3) N = 39

THE ALPHA VECTOR=

```
-6.599996E-03  4.887999E-01  -1.751199E+00  2.260499E+00  6.599996E-03  -1.244999E-01
 2.890000E-02  9.749997E-02   7.271961E+15
```

(4) N = 49

THE ALPHA VECTOR=

```
-6.600000E-03  4.888000E-01  -1.751200E+00  2.260500E+00  6.600000E-03  -1.245000E-01
 2.890000E-02  9.749997E-02  -1.482148E-14
```

(5) N = 59

THE ALPHA VECTOR=

−6.600000E−03	4.888000E−01	−1.751200E+00	2.260500E+00	6.600000E−03	−1.245000E−01
2.890000E−02	9.749997E−02	−6.761258E−14			

From these results, the estimation errors S_A and S_B are acceptable for most analysis/design studies using linear theory.

5-5 Concluding remarks

In this chapter we have developed a method to identify the system order and parameters of discrete-time systems, the order determination is based on the rank difference between two appropriately constructed matrices. The basic procedure employed to identify the unknown parameters is based on a special output data vector $\underline{y}$ (K ; N) linear combination of the input -output measurement data.

The method requires the computations of only one matrix inversion. The order determined may be incorrect when unsuitable input sequences were used. In fact, since sampling length N is finite, there is always a possibility that the actual order of the system is higher than the one determined.

PART- II

In recent years, the revolutionary advancements in the technology of microprocessors have enhanced the importance of digital control systems. The advantages of digital control systems include : increased reliability, improved sensitivity, decreased size and weight, and time-sharing capabilities. The more distinct advantages of digital control systems are the increased flexibility of the control algorithms and the complex decision making capability of the processor.

This part of the thesis is concerned with some of the fundamental aspects of designing digital controllers for time-invariant linear systems. The two chapters in this part are self-contained and aimed at presenting an outline of two design techniques which may follow the identification process described in Part-I.

This part has the following chapters :

Chapter 6 : A mircoprocessor-based output deadbeat controller.

Chapter 7 : Design of digital systems.

Chapter – 6 A MICROPROCESSOR-BASED OUTPUT DEADBEAT CONTROLLER

The recent advancements in the technology of microprocessors and microelectronics have drastically changed the philosophy of the design and implementation of controllers. With the advent of inexpensive microprocessors, the digital control algorithms originally developed for the aerospace and advanced process control industries, can now be implemented even for basic applications [119 – 120].

For the industrial control systems requiring a finite settling time and zero steady-state error to a unit step input, the output deadbeat control design can now be accomplished [120 – 121]. The conventional output deadbeat control system with digital cascade compensation is shown in Fig. 6 – 1 [119 – 124]. Recently a systematic deadbeat control algorithm has been presented in the form of feedback and pre-compensation reduced parameter sensitivity [Fig. 6 – 2 (125)].

In this chapter, a self-contained hardware system, based on the Zilog Z-80A microprocessor is developed for the implementation of the deadbeat output compensation system shown in Fig. 6 — 2. Implementation of the more widely used industrial digital lead-lag compensator and the digital PID controller is also discussed. The algorithm developed has been used as the basis for the control software on a Z-80 A based custom-designed controller.

6 — 1 Deadbeat output compensation algorithm

The deadbeat output compensation system [125] shown in Fig. 6 — 2 is obtained by applying a linear state transformation technique to the deadbeat state compensation system as shown in Fig. 6 — 3.

In Fig. 6 — 3, the discrete state equation of the continuous plant G(s) can be expressed as*

$$X_c(k+1) = A_c X_c(k) + B_c U_c(k) \qquad (6-1)$$

where A_c is $n \times n$, and B_c is $n \times 1$. A digital compensator D(z), with the general form shown in (6 — 2).

$$D(z) = \frac{U_c(z)}{U_d(z)} = \frac{q_0 z + q_1}{z + p_1} \qquad (6-2)$$

can be described as

$$X_d(k+1) = A_d X_d(k) + B_d U_d(k)$$

* capital letters indicate vectors and matrices, lower case letters indicate scalars.

$$A_d = -p_1 \tag{6-4}$$

$$B_d = q_1 - q_0 p_1 \tag{6-5}$$

The control law $u_c(k)$ is

$$u_c(k) = C_d\, x_d(k) + D_d\, u_d(k) \tag{6-6}$$

with

$$C_d = 1 \tag{6-7}$$

$$D_d = q_0 \tag{6-8}$$

while the compensator input $u_d(k)$ is

$$u_d(k) = r(k) - H'\, x_c(k) \tag{6-9}$$

where H is n x 1, and H' is the transpose of H.

The dynamic equation of the closed-loop system can then be described as

$$\begin{bmatrix} x_c\,(k+1) \\ \hline x_d\,(k+1) \end{bmatrix} = \left[\begin{array}{c|c} A_c - D_d B_c H' & C_d B_c \\ \hline -B_d H' & A_d \end{array}\right] \begin{bmatrix} x_c(k) \\ \hline x_d(k) \end{bmatrix} + \begin{bmatrix} D_d B_c \\ \hline B_d \end{bmatrix} r(k) \tag{6-10}$$

$$y(k) = C' \begin{bmatrix} x_c(k) \\ \hline x_d(k) \end{bmatrix} \tag{6-11}$$

where C' is (n + 1) × 1, and C' is the transpose of C.

Thus the digital transfer function of the closed-loop system is

$$\frac{Y(z)}{R(z)} = C' \left[zI - \left[\begin{array}{c|c} A_c - D_d B_c H' & C_d B_c \\ \hline -B_d H' & A_d \end{array}\right] \right]^{-1} \begin{bmatrix} D_d B_c \\ \hline B_d \end{bmatrix} \tag{6-12}$$

with the linear transformation

$$x_t(k) = x_d(k) + T' x_c(k) \qquad (6-13)$$

where T is n x 1, and T' is the transpose of T , and , have

$$x_t(k+1) = x_d(k+1) + T' x_c(k+1) \qquad (6-14)$$

Substituting (6 – 1) and (6 – 3) into (6 – 14), and using (6 – 9) and (6 – 13), we have

$$x_t(k+1) = A_d x_t(k) + (T'A_c - A_d T' - B_d H') x_c(k) + T' B_c u_c(k) + B_d r(k) \qquad (6-15)$$

Similarly, using (6 – 9) and (6 – 13) in (6 – 16) from (6 – 6) .

$$u_c(k) = C_d x_t(k) + (-C_d T' - D_d H') x_c(k) + D_d r(k) \qquad (6-16)$$

In order to select a linear transformation T such that both $x_t(k+1)$ in (6 – 15) and $u_c(k)$ in (6 – 16) depend on the output response $y(k)$ instead of all the states $x_c(k)$, we must have

$$(T'A_c - A_d T' - B_d H') x_c(k) = L C' x_c(k) = L y(k) \qquad (6-17)$$

$$(-C_d T' - D_d H') x_c(k) = M C' x_c(k) = M y(k) \qquad (6-18)$$

where C is n x 1, C' is the transpose of C, while L and M are constants.

Based on the above discussion, the control law and the transformed digital compensator can be described, respectively, as

$$u_c(k) = C_d x_t(k) + M y(k) + D_d r(k) \tag{6-19}$$

$$x_t(k+1) = D x_t(k) + E y(k) + W r(k) \tag{6-20}$$

where

$$D = A_d + C_d T' B_c \tag{6-21}$$

$$E = L + M T' B_c \tag{6-22}$$

$$W = B_d + D_d T' B_c \tag{6-23}$$

D, E, and W are all constants. Once we find a linear transformation T and the scalars L and M to satisfy (6-24) and (6-25).

$$T' A_c - A_d T' - B_d H' = LC' \tag{6-24}$$

$$-C_d T' - D_d H' = MC' \tag{6-25}$$

the deadbeat solution of (6-19) and (6-20) can be obtained by using the deadbeat model [125].

$$M(z) = \frac{Y(z)}{R(z)} = a_1 z^{-1} + \sum_{i=2}^{k-1} (a_i - a_{i-1}) z^{-i}$$

$$+ (1 - a_{k-1}) z^{-k} \tag{6-26}$$

In the above model a_i represents the amplitude (ith sample) of the unit step response (Fig. 6-4) for the closed-loop deadbeat system (6-12). Thus if the system order is k, then

$$C' \left[zI - \left[\begin{array}{c|c} A_c - D_d B_c H' & C_d B_c \\ \hline -B_d H' & A_d \end{array} \right] \right]^{-1} \left[\begin{array}{c} D_d B_c \\ \hline B_d \end{array} \right]$$

$$= a_1 z^{-1} + \sum_{i=2}^{k-1} (a_i - a_{i-1}) z^{-i}$$

$$+ (1 - a_{k-1}) z^{-k} \qquad (6-25)$$

The deadbeat state compensation design obtained from (6—27) can then be substituted into (6—24) and (6—25) to solve T, L and M for the purpose of calculating D, E and W from (6—21), (6—22) and (6—23). The deadbeat solution of (6—19) and (6—20) can thus be obtained.

6—2 Hardware system and implementation *

A purpose-built hardware system for the microprocessor-based deadbeat controller is presented in this section. Fig. 6—5, shows the schematic diagram of this hardware system, which is developed around Zilog Z-80A microprocessor.

The hardware system depicted in Fig. 6—5 has a universal configuration, because it can be used not only for the implementation of the output deadbeat control algorithm described above but also for the implementation of the widely used digital lead-lag compensation and the digital PID control algorithms.

The sample-and-holds circuits (S/H) shown in Fig. 6—5 are

* Assistance of Dr. F. Y. Shih in the hardware design and construction is gratefully acknowledged.

used to take the samples of the reference input r(t) and the output response y(t) at each sampling instant, and then to hold the sampled values over the corresponding sampling periods for analog-to-digital conversion. The sampling frequency generator (SFG) provides the sampling command for the sample-and-holds and the acknowledging signal for the microprocessor.

The two-input multiplexer (MPX) is used to provide the simultaneous sample-and-hold operation of the hardware system depicted in Fig. 6—5. The analog-to-digital converter (ADC) converts the signal coming from the output of the multiplexer into a corresponding digital code for the microprocessor.

The microprocessor processes the digital data in accordance with the control algorithms implemented in ROM, and coordinates the interfacing devices such as (MPX), (ADC), and (DAC). The digital-to-analog converter (DAC) converts the digital code of the control signal $u_c(k)$ coming from the peripheral input-output device (PIO) into a corresponding analog value and holds it until the next control signal is ready.

In order to increase the computational accuracy, 32-bit floating point format was used. In this format, the most significant byte is a sign byte, with 80H representing the positive values and 00H representing the negative values. The second byte is the high byte of the mantissa, while the third byte is the low byte of the mantissa. The fourth byte represents the exponent. The binary point is assumed at the leftmost end of this format.

Since the operating values in the hardware system are confined to be in the range ± 5.0 (00 to FF hexadecimal), the exponent byte can be set as 03H. Thus the floating point numbers are in the range $\pm 0.625 \times 2^3$. A flowchart for the digital controller with deadbeat output compensation algorithm is shown in Fig.6−6.

In the flowchart shown in Fig.6−6, the initialization routine initializes the (PIO), (RAM), and the data aquisition devices. The binary to floating point conversion routine converts the binary values coming from (ADC) into the 32-bit floating point format described above. The floating point multiplication, addition, and subtraction routines are used to manipulate the floating point arithmetic operations.

An exponent normalization subprogram is utilized to equalize the exponents of the floating point values in the operation of addition or subtraction. Another subprogram incorporated in the floating point arithmetic routines is the overflow subroutine for treating the overflow data.

Once the 32-bit floating point control values $u_c(k)$ are obtained, the floating point to binary conversion routine converts them into the corresponding binary values for (DAC) conversion.

The program length is slightly more than 0.7 kbytes of (ROM). In addition, 40 bytes of (RAM) are required for scratch pad of intermediate values. The execution time are found to be less than 8 mseconds.

6−3 Lead-lag compensator and PID controller

The microprocessor-based implementation of the widely used digital lead-lag compensator and PID controller is discussed in this section. Without any modification, the hardware system shown in Fig. 6−5 can be used to implement these two control algorithms.

(a) Digital lead-lag compensator

In the conventional digital control system shown in Fig. 6−1, the transfer function of a typical digital lead-lag compensator can be expressed as

$$D(z) = \frac{U_c(z)}{U_d(z)} = \frac{Q_0 + Q_1 z^{-1}}{1 + p_1 z^{-1}} \qquad (6-28)$$

Based on (6−28), the difference equation shown below can be obtained.

$$u_c(k) = Q_0 u_d(k) + Q_1 u_d(k-1) - p_1 u_c(k-1) \qquad (6-29)$$

From Fig.6−1,

$$u_d(k) = r(k) - y(k) \qquad (6-30)$$

Equations (6−29)and(6−30)constitute the control algorithm of the digital lead-lag compensator.

The flowchart of the microprocessor-based digital lead-lag compensator is depicted in Fig.6−7, Looking at Fig.6−7, we find that the operation of the digital lead-lag compensator is pretty similar to that of deadbeat controller with output compensation algorithm. The only existing difference between Fig.6−6 and Fig.6−7 is the discrete equations manipulated by them.

The program requires less than 0.67 K bytes of (ROM) and 40 bytes of (RAM). The total execution time is less than 6 msecs.

(b) Digital PID Controller

From Fig.6−1, the transfer function of the typical digital PID controller may be derived as

$$D(z) = \frac{U_c(z)}{U_d(z)} = K_p + \frac{K_I T}{2}\,\frac{z+1}{z-1} + \frac{K_D}{T}\,\frac{z-1}{z} \qquad (6-31)$$

Equation (6−31) can be expressed as

$$D(z) = \frac{U_c(z)}{U_d(z)} = \frac{Q_0 + Q_1 z^{-1} + Q_2 z^{-2}}{1 - z^{-1}} \qquad (6-32)$$

with

$$Q_0 = K_p + \frac{K_I T}{2} + \frac{K_D}{T}$$

$$Q_1 = -K_p + \frac{K_I T}{2} - \frac{2K_D}{T} \qquad (6-33)$$

$$Q_2 = \frac{K_D}{T}$$

Based on (6−32), the difference equation shown in (6−34) can be obtained.

$$u_c(k) = Q_0 u_d(k) + Q_1 u_d(k-1) + Q_2 u_d(k-2) + u_c(k-1) \qquad (6-34)$$

From Fig. 6−1, it is apparent that (6−30) is also required for the case of digital PID controller. Thus, (6−34) and (6−30) constitute the control algorithm of the digital PID controller.

The flow-chart of the microprocessor-based digital PID controller is shown in Fig. 6−8. This shows that the operation of the digital PID controller is similar to that of deadbeat controller with output compensation algorithm. The only difference between Fig. 6−6 and Fig. 6−8 being the numerical equations.

In addition, the digital PID control law shown in (6−34) and the control law of the digital lead-lag compensator, shown in (6−29), are similar to each other. But, in (6−34), the term $u_d(k-2)$ is incorporated and the coefficients Q_0, Q_1, and Q_2 must be calculated in accordance with (6 — 33) by using the values K_P , K_I , K_D, and T.

The program length was less than 0.95 Kbytes of (ROM) and 60 bytes of (RAM). The total execution time was less than 10 mseconds.

The performance tests of an illustrative examples are briefly discussed : In Fig. 6—2 , for a plant

$$G(s) = \frac{70}{s\ (\ s + 3.5\ )}$$

The parameters of the deadbeat controller using the above procedure are $D = -0.651$, $E = 3.855$, $W = 0.545$, $M = -3.365$, $D_d = 0.7$ and $C_d = 1.0$.

The output response y(t) for a unit step input $U_c(t)$ of the system with the above deadbeat controller is shown in Fig. 6—9. This demonstrates the effectiveness of the controller. The response in Fig. 6—10 shows the ability of the controller to cope with an input disturbance.

6 — 4 Concluding remarks

In this chapter, a microprocessor-based output deadbeat controller for digital servo system with finite settling time has been presented. A purpose-built hardware has been used to implement this control algorithm.

A fairly simple system has been used to illustrate the general validity of the controller design methodology.

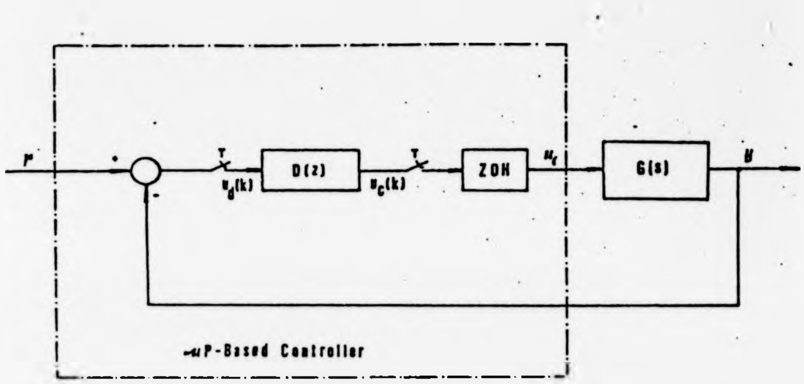

Fig. 6—1 Conventional Output Deadbeat Control System

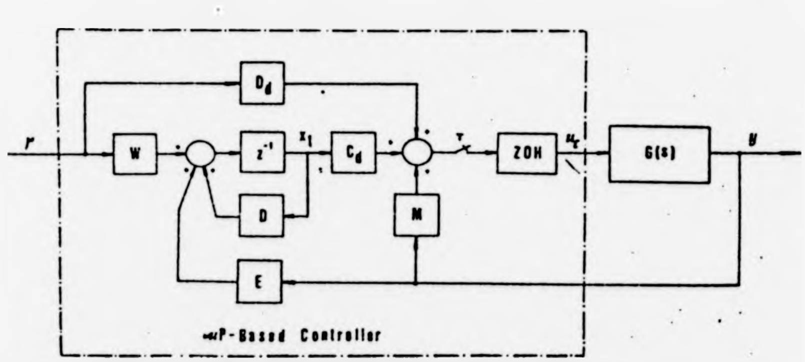

Fig. 6—2 Deadbeat Output Compensation System

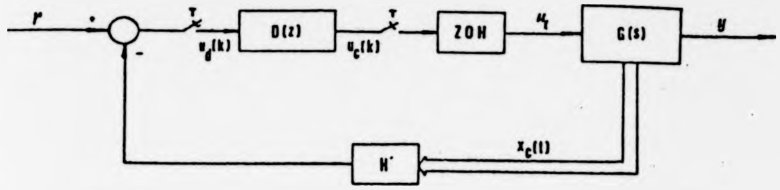

Fig. 6—3 Deadbeat State Compensation System

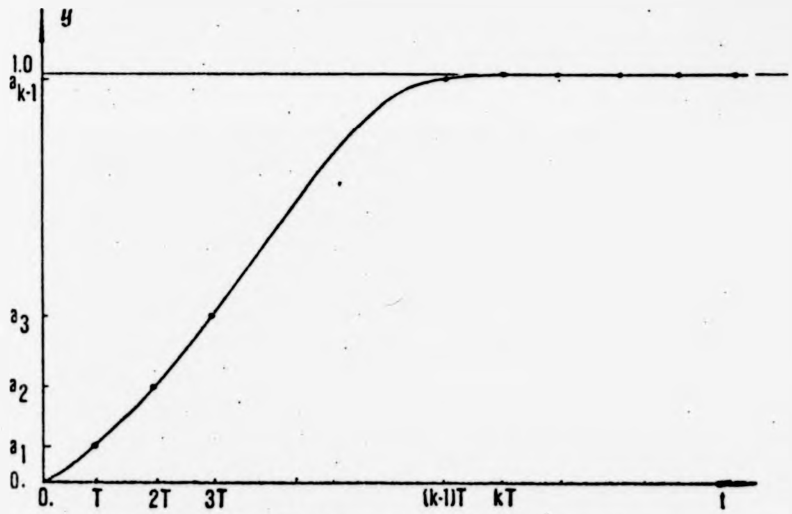

Fig. 6—4 Typical Deadbeat Unit Step Response

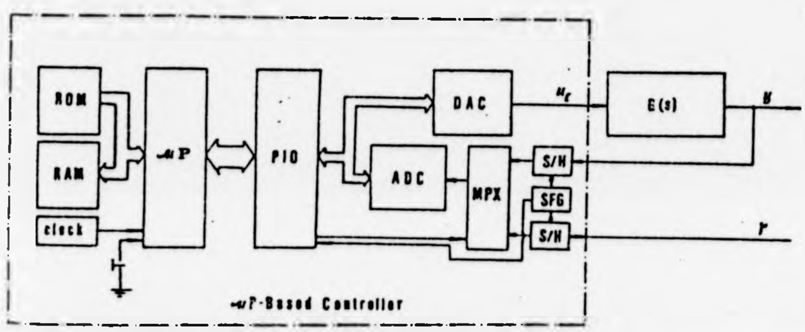

Fig. 6—5 Hardware System of the Microprocessor-Based Deadbeat Controller

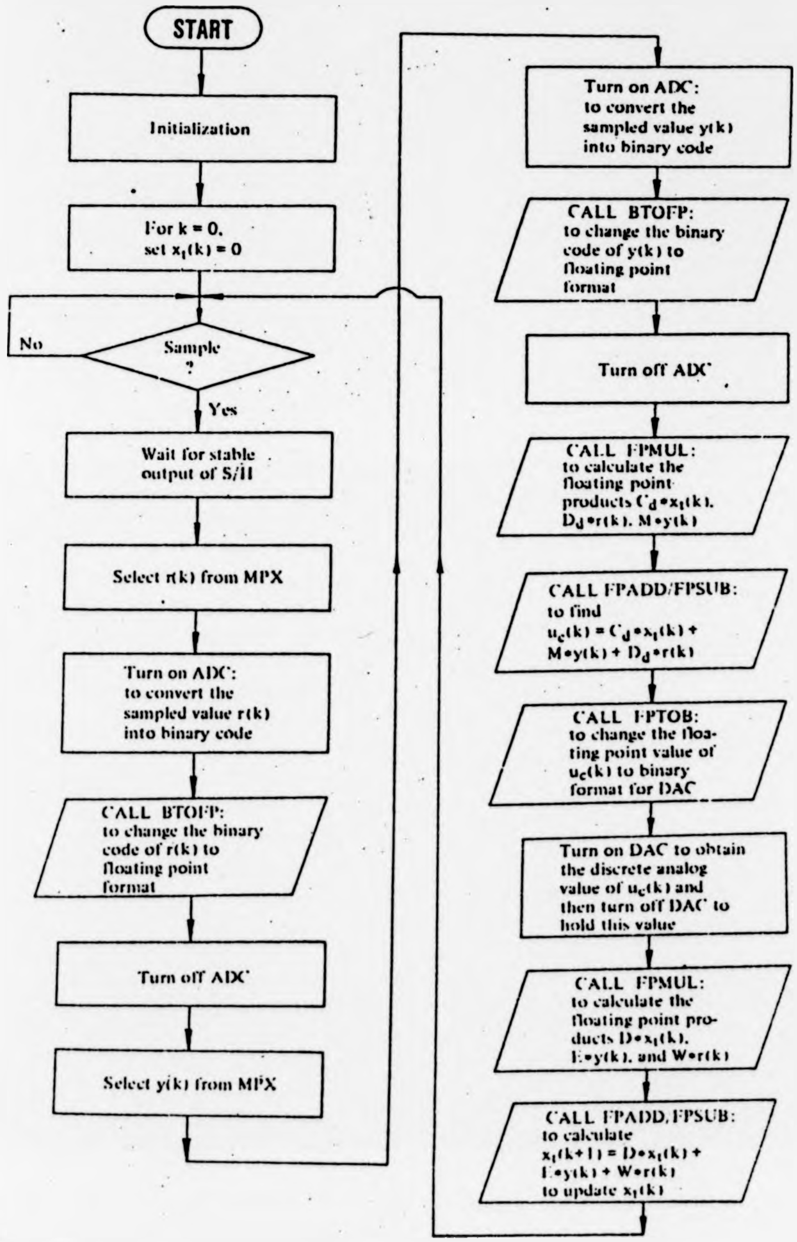

Fig. 6—6 Flowchart of the Digital Controller with Deadbeat Output Compensation Algorithm

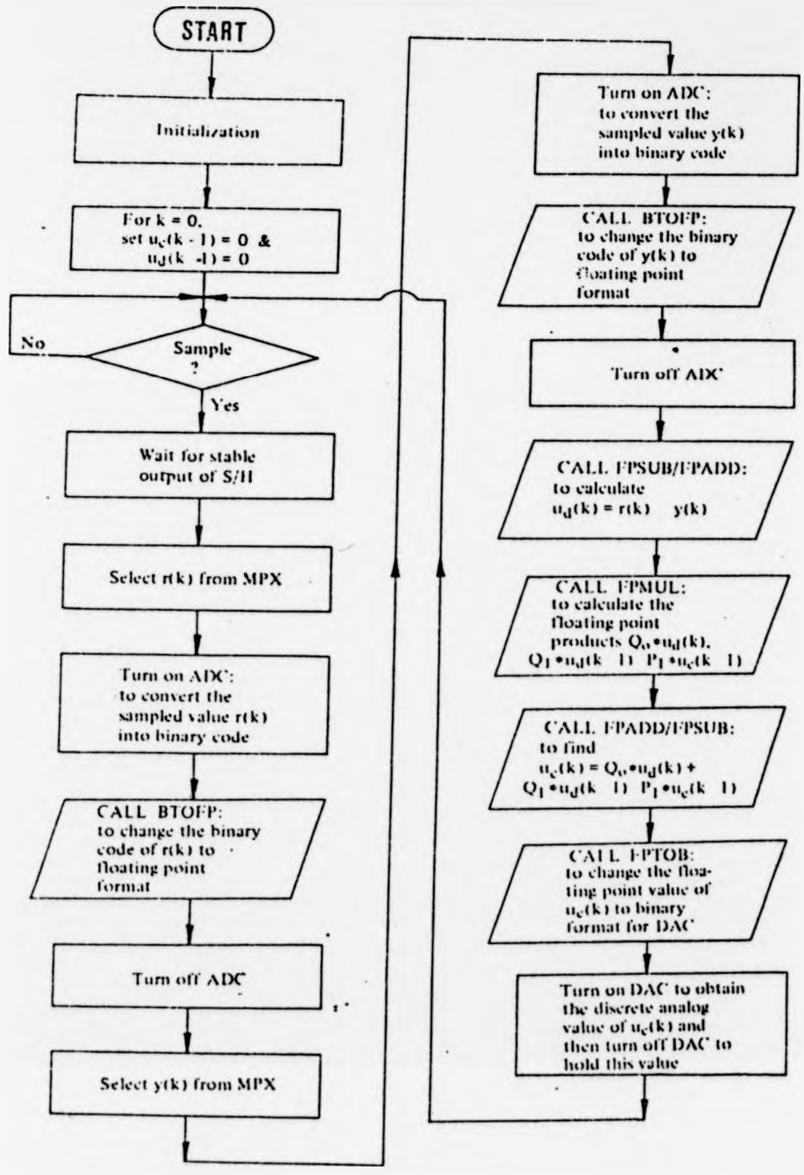

Fig. 6–7 Flowchart of the Digital Controller with Lead-Lag Compensation Algorithm

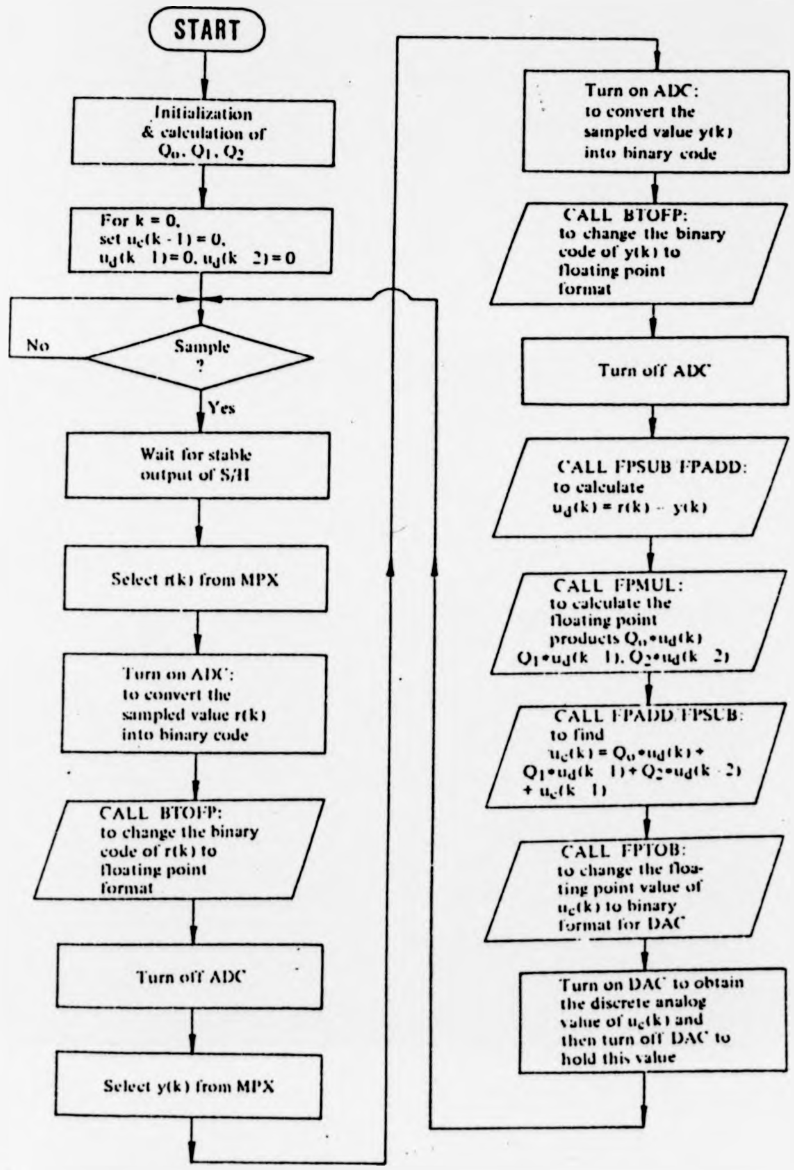

Fig. 6–8 Flowchart of the Digital Controller with PID Control Algorithm

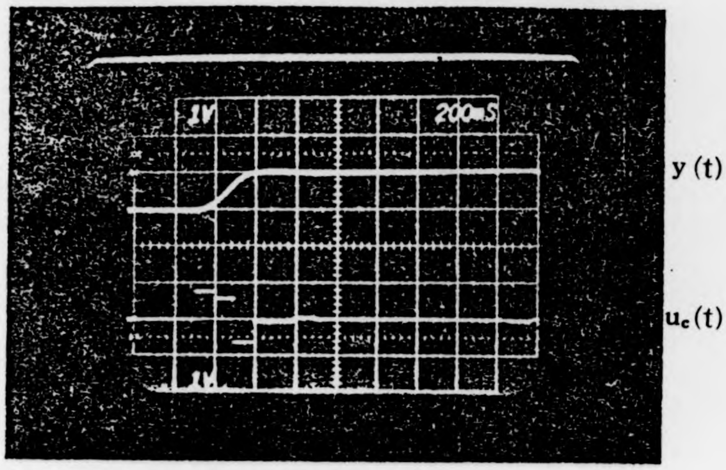

Fig. 6—9 Unit Step Response and Control Input

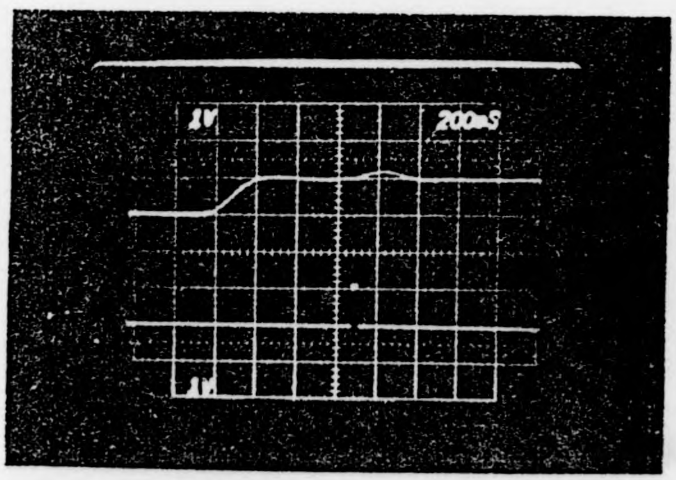

Fig. 6—10 The Performance and Disturbance Signal

Chapter — 7 DESIGN OF DIGITAL SYSTEMS

The conventional design methods employ the root-locus or Bode diagram techniques to synthesize the digital controllers [114-116]. These techniques, even with the aid of digital computers for the determination of controller parameters, are rather slow. This is due mainly to the trial-and-error nature, with only one of the several design parameters being treated at a time. The iterations to study the effects of several parameters can become tedious.

In this chapter a computer aided design technique is developed. This technique is based on the parameter optimization approach using the Quasi-Newton nonlinear programming algorithm [117].

A performance index PI, which consists of three terms for penalizing the system error, the actuating force, and the location of the closed-loop poles respectively, is formulated. The weighting factors are used to change the emphasis on the individual terms of the performance index PI.

In a straightforward manner, practical constraints can be placed upon the allowable regions of the design parameters of the digital controllers. The optimized design parameters can be determined when the performance index PI is minimized.

A computer aided design program has been implemented in FORTRAN such that a single computer run derive the optimized design parameters of the digital controllers as well as the unit step response of the overall digital control system; design example is used to demonstrate the application of this systematic CAD technique to a 10th-order digital flight control system.

7—1 Digital control systems

A general digital control system is as depicted in Fig. 7—1 The state equation of the continuous plant is

$$\dot{\underline{x}}_c(t) = F\,\underline{x}_c(t) + G\,u_c(t) \tag{7—1}$$

where F is $n_c \times n_c$, and G is $n_c \times 1$.

If

$$A_c = e^{FT} \tag{7—2}$$

$$B_c = \int_0^T e^{F\partial} G \, d\partial \qquad (7-3)$$

then the discretized state equation is

$$\underline{x}_c(k+1) = A_c \, \underline{x}_c(k) + B_c \, u_c(k) \qquad (7-4)$$

where A_c is $n_c \times n_c$, and B_c is $n_c \times 1$.

The state equation of the digital controller is

$$\underline{x}_d(k+1) = A_d \, \underline{x}_d(k) + B_d \, \underline{u}_d(k) \qquad (7-5)$$

where A_d is $n_d \times n_d$, and B_d is $n_d \times m_d$.

The actuating force of the controlled plant is assumed to be

$$u_c(k) = C_d \, \underline{x}_d(k) + D_d \, \underline{u}_d(k) \qquad (7-6)$$

where C_d is $1 \times n_d$, and D_d is $1 \times m_d$.

The input signal of the digital controller is

$$\underline{u}_d(k) = C_c \, \underline{x}_c(k) + B_r \, r(k) \qquad (7-7)$$

where C_c is $m_d \times n_c$, and B_r is $m_d \times 1$.

Then the state equation of the overall system becomes

$$\underline{x}(k+1) = A \, \underline{x}(k) + B \, r(k) \qquad (7-8)$$

where

$$A = \left[\begin{array}{c|c} A_c + B_c D_d C_c & B_c C_d \\ \hline B_d C_c & A_d \end{array} \right] \qquad (7-9)$$

$$B = \left[\frac{B_c D_d B_r}{B_d B_r}\right] \qquad (7-10)$$

$$\underline{x}(k) = \left[\frac{\underline{x}_c(k)}{\underline{x}_d(k)}\right] \qquad (7-11)$$

The output equation is

$$y(k) = C^T \underline{x}(k) \qquad (7-12)$$

where C is $(n_c + n_d) \times 1$, and C^T is the transpose of C.

Given $\underline{x}(o)$ and $u(k)$ over the interval $[o, k-1]$, the expression for $\underline{x}(k)$ at any $k>o$ can be derived as follows: Substituting successively $k = o, 1, \cdots, k-1$ in (7−8), we obtain

$$\underline{x}(k) = A^k \underline{x}(o) + \sum_{j=0}^{k-1} A^j B\, r(k-j-1) \qquad (7-13)$$

where $A^k = \phi(k)$ is the state transition matrix.

Thus (7−13) can be expressed as

$$\underline{x}(k) = \phi(k)\, \underline{x}(o) + \sum_{j=0}^{k-1} \phi(j)\, B\, r(k-j-1) \qquad (7-14)$$

For the case of zero initial conditions in (7−14), i.e. $\underline{x}(o) = 0$, the output equation (7−12) can be expressed as

$$\begin{aligned} y(k) &= \sum_{j=0}^{k-1} C^T \phi(j)\, B\, r(k-j-1) \\ &= \sum_{m=0}^{k-1} C^T \phi(k-m-1)\, B\, r(m) \qquad (7-15) \end{aligned}$$

7 — 2 Performance index PI

A performance index PI with the form shown in (7−16) is formulated in this section.

$$PI = WF_1 \times PI_1 + WF_2 \times PI_2 + WF_3 \times PI_3$$

$$= \sum_{i=1}^{3} WF_i \times PI_i \qquad (7-16)$$

where WF_i are the weighting factors, and PI_i are the performance indices as follows :

$$PI_1 = \sum_{k=0}^{N} \{ r(k) - y(k) \}^2 \qquad (7-17)$$

where $r(k)$ is the reference input, $y(k)$ is the output response, and $r(k) - y(k)$ is the error function of the closed-loop system.

$$PI_2 = \sum_{k=0}^{N} u(k)^2 \qquad (7-18)$$

where $u(k)$ is the actuating force to the controlled plant.

$$PI_3 = \sum_{i=1}^{n} \{ ZR(i)^2 + ZI(i)^2 \}^{1/2} \qquad (7-19)$$

where $ZR(i)$ is the real part of the closed-loop poles, and $ZI(i)$ is the imaginary part of the closed-loop poles, and n is the order of the overall system.

(a) PI_1 shown in (7−17) is used to penalize the system error, and can be computed at each sampling instant by solving the state equation (7−8) iteratively from $k=0$ to $k=N$, where N is the number of sampling instants to be evaluated. Equation (7−15) is a typical output solution to be used in the evaluation of PI_1.

(b) PI_2 shown in (7−18) is used to penalize the actuating force, which can be represented by the linear combination of some state variables as defined in (7−8) and (7−14) and /or reference input. Thus the evaluation of the actuating force $u(k)$ at the k-th sampling instant is the same as the evaluation of the linear combination of the state variables and/or the reference input at the k-th sampling instant.

(c) PI_3 shown in (7−19) is used to penalize the location of the closed-loop poles, which are the same as eigenvalues of the closed-loop state matrix. In order to obtain the finite settling time design, the closed-loop poles in the z-plane should be placed as close to the origin as possible [118−119].

7 — 3 Computational techniques

In this section, a parameter optimization program that uses the Quasi-Newton nonlinear programming algorithm to minimize the performance index of N variables, as specified in (7−20), is introduced. Let the general performance be expressed as:

$$PI = F(p_1, p_2, \cdots, p_n) \qquad (7-20)$$

The iteration steps of the Quasi-Newton nonlinear programming algorithm is summarized as follows :

1. Assuming a minimization algorithm, we start with a positive definite matrix H_0 and some initial point $\underline{p}^0$. For convenience, H_0 can be chosen to be the identity matrix I.

2. The general iteration step begins here. We designate this the k-th iteration. Calculate the gradient vector $\nabla F(P^k)$

3. Calculate a direction in which to move. This is given by $s^k = -H_k \nabla F(\underline{p}^k)$

4. In order to move in the direction s^k, we need to calculate a step size ∂^k. Hence, we wish to choose ∂ so as to minimize $F(\underline{p}^k + \partial s^k)$. Formally, $\partial^k = \min F(\underline{p}^k + \partial s^k)$

5. Calculate the descent step $\underline{p}^{k+1} = \underline{p}^k + \partial^k s^k$

6. Calculate $y^k = \nabla F(\underline{p}^{k+1}) - \nabla F(\underline{p}^k)$

7. Calculate the next approximation in the sequence of Hessian matrices H_{k+1}

8. Terminate the calculation if $F(\underline{p}^k) - F(\underline{p}^{k+1}) \leqslant \epsilon$. If $F(\underline{p}^k) - F(\underline{p}^{k+1}) > \epsilon$, return to step 2 using H_{k+1} as the new H_k.

Generally, the computational procedure for the design of digital control systems includes three major steps. Firstly, the continuous plant is discretized, and the discrete state equation and the transfer function of the overall closed-loop system is constructed. Secondly, the desired parameters are optimized by means of minimizing the specified performance index. Finally, the optimized system is evaluated through a digital simulation.

For the parameter optimization problem,

$$\min PI = \min F(p_1, p_2, \cdots, p_n) \qquad (7\text{—}21)$$

subject to the constraints

$$A_i \leqslant p_i \leqslant B_i \quad , \quad i = 1, 2, \cdots, n \qquad (7\text{—}22)$$

an effective method of constraining the independent variables is to apply the transformation as shown in (7—23).

$$p_i = A_i + (B_i - A_i)\ P(v_i) \qquad (7\text{—}23)$$

where

$$P(v_i) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{v_i} \exp(-t^2)\ dt = 0.5(1.0 + erf(v_i)) \qquad (7\text{—}24)$$

Specifically, the parameter optimization technique for the design of digital control systems consists of

1. Initialization of A_i and B_i, $i = 1, 2, \cdots, n$, to the constraint limits, as shown in (7—22).

2. Initialization of the parameters p_i, $i = 1, 2, \cdots, n$, to values that lie between A_i and B_i.

3. Transfer of the bounded parameters p_i to the unbounded parameters v_i, as shown in (7—23).

4. Construction of the closed-loop digital control system, as shown in (7—8) through (7—12).

5. Evaluation of performance index, as shown in (7—16) through (7—19).

6. Minimization of the performance index via Quasi-Newton nonlinear programming algorithm.

7. Transfer of the unconstrained optimum parameters v_i back to the constrained ones p_i for $i = 1, 2, \cdots N$.

8. Evaluation of the optimized system response.

In addition to the explicit parameter constraint, as shown in (7—22), an implicit constraint is applied to insure that all closed-loop poles are located inside the unit circle.

During the iteration, the convergence criteria is used to insure that all the poles are inside the unit circle. If a violation occurs, the boundary limit of the desired parameters is decreased by a specific value such that the new parameter boundary is smaller than the current one.

The stopping conditions are given through two input parameters, one specifies the number of significant digits of accuracy desired, the other specifies the maximum number of function evaluations allowed during the iteration process.

When the iteration terminates, the final value of the performance index is checked. If the value is less than a specified value, the procedure is concluded. Otherwise, a new initial guess for parameters is set and the optimization routine is repeated. The maximum number of times that the optimization routine can be called, with different initial guesses, is preset through an input parameter.

The flowchart illustrating the CAD procedure is as shown in Fig. 7-2.

7-4 Design example

Consider the digital flight control system as Fig. 7-3. The continuous state equation, to describe the aircraft, the actuator and the filter, is

$$\dot{\underline{x}}_c(t) = F\,\underline{x}_c(t) + G\,u_c(t) \tag{7-25}$$

with

$$F = \begin{bmatrix} -0.0486 & -0.0371 & -30.923 & 0 & -21.6689 & 0 & 0 & 0 \\ -0.1063 & -0.5436 & 8.8845 & 593.133 & -139.63 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0.0001 & 0.0004 & 0 & -0.49 & -2.2882 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1066.6756 & -37.2324 & 1066.6756 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -900 & -42 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 900 \end{bmatrix}$$

The parameters to be optimized are the gain and time constants of the compensators $G_1(s)$ and $G_2(s)$ as shown in (7-26) and (7-27)

$$G_1(s) = G\frac{T_1 s + 1}{T_2 s + 1} \tag{7-26}$$

$$G_2(s) = \frac{1}{T_3 s + 1} \tag{7-27}$$

Define the desired design parameters as

$$R_1 = T_1, \quad R_2 = T_2, \quad R_3 = G, \quad R_4 = T_3 \tag{7-28}$$

Applying Tustin operator on (7-26) and (7-27), and using the parameters defined in (7-28), we have the discrete compensators $G_1(z)$ and $G_2(z)$ as

$$G_1(z) = R_3\frac{R_1 + \frac{1}{2}T\frac{z+1}{z-1}}{R_2 + \frac{1}{2}T\frac{z+1}{z-1}}$$

$$= \frac{R_3\frac{R_1+\frac{1}{2}T}{R_2+\frac{1}{2}T}z + R_3\frac{\frac{1}{2}T - R_1}{\frac{1}{2}T + R_2}}{z - \frac{R_2 - \frac{1}{2}T}{R_2 + \frac{1}{2}T}} \tag{7-29}$$

$$G_2(z)=\frac{\frac{1}{2}T\frac{z+1}{z-1}}{R_4+\frac{1}{2}T\frac{z+1}{z-1}}$$

$$=\frac{\frac{\frac{1}{2}T}{R_4+\frac{1}{2}T}\,z+\frac{\frac{1}{2}T}{R_4+\frac{1}{2}T}}{z-\frac{R_4-\frac{1}{2}T}{R_4+\frac{1}{2}T}} \qquad (7\text{—}30)$$

By using the notations shown in the previous sections, (7—29) and (7—30) can be represented as

$$\underline{x}_d(k+1)=A_d\,\underline{x}_d(k)+B_d\,\underline{u}_d(k)$$

$$=\begin{bmatrix}A_d(1,1) & 0\\ 0 & A_d(2,2)\end{bmatrix}\underline{x}_d(k)+\begin{bmatrix}B_d(1,1) & 0\\ 0 & B_d(2,2)\end{bmatrix}\underline{u}_d(k)$$

(7—31)

$$y_{d_1}=-x_{d_1}(k)+D_d(1,1)\ u_{d_1}(k) \qquad (7\text{—}32)$$

$$y_{d_2}=x_{d_2}(k)+D_d(1,2)\ u_{d_2}(k) \qquad (7\text{—}33)$$

$$u_c(k)=y_{d_1}(k)+y_{d_2}(k)$$

$$=[-1\quad 1]\ \underline{x}_d(k)+[D_d(1,1)\ \ D_d(1,2)]\underline{u}_d(k)$$

$$=C_d\ \underline{x}_d(k)+D_d\ \underline{u}_d(k) \qquad (7\text{—}34)$$

with

$$A_d(1,1)=(R_2-\tfrac{1}{2}T)/(\tfrac{1}{2}T+R_2)$$

$$A_d(2,2)=(R_4-\tfrac{1}{2}T)/(\tfrac{1}{2}T+R_4)$$

$$D_d(1,1)=R_3(\tfrac{1}{2}T+R_1)/(\tfrac{1}{2}T+R_2)$$

$$D_d(1,2) = \tfrac{1}{2}T / (\tfrac{1}{2}T + R_4) \qquad (7\text{-}35)$$

$$B_d(1,1) = A_d(1,1)\,D_d(1,1) + R_3(\tfrac{1}{2}T - R_1)/(\tfrac{1}{2}T + R_2)$$

$$B_d(2,2) = A_d(2,2)\,D_d(1,2) + \tfrac{1}{2}T / (\tfrac{1}{2}T + R_4)$$

where $\underline{x}_d(k) = [\, x_{d_1}(k)\ \ x_{d_2}(k)]^T$, and $\underline{u}_d(k) = [\, u_{d_1}(k)\ \ u_{d_2}(k)]^T$.
The sampling time $T=0.04$ sec for this system.
The control vector to the digital compensators is

$$\underline{u}_d(k) = C_c\, \underline{x}_c(k) + b_r\, r(k)$$
$$= \begin{bmatrix} 0 & 0 & 0 & \partial & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \underline{x}_c(k) + \begin{bmatrix} -1 \\ 0 \end{bmatrix} r(k) \qquad (7\text{-}36)$$

where $\partial = 1$ for the closed-loop system, whereas $\partial = 0$ for the open-loop system.
The output matrix is

$$C^T = [\,0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\,] \qquad (7\text{-}37)$$

The design conditions and the parameters designed through the optimization procedure are shown in Table 7-1 . The output response of the parameter optimized digital flight control system is shown in Fig. 7-4 .

Table 7-1 Design Conditions and Result of Design Example

```
THE NUMBER OF DESIRED PARAMETERS N = 4
THE BOUNDARY VALUE OF DESIRED PARAMETERS

DA =  5.000D-02    5.000D-02    5.000D-02    5.000D-02
DB =  5.000D+00    5.000D+00    5.000D+00    5.000+00

THE PRESENT VALUE OF WEIGHTING FACTOR:
WF(1) =  0.500D+01   WF(2) =  0.100D+01   WF(3) =  0.300D+00
THE STOPPING CRITERIA:
THE NO. OF ACCURACY DIGITAL: NSIG = 3
MAX. NO. OF EVALUATIONS: MAXFN = 300
THE MAX. VALUE OF P.I.: TH = 0.10000D+04
THE STEP SIZE FOR BOUNDARY CHANGED: TD = 0.10000D+01
OUTPUT INFORMATION:
NER = 0  : NER = 0 IF ALL POLES INSIDE UNIT CIRCLE
THE PERFORMANCE INDEX EVALUATION:
PI(1) = 0.35857D+01   PI(2) = 0.24803D+02   PI(3) = 0.68086D+01
PERFORMANCE INDEX: PI = 0.32177999D+02
THE OPTIMIZED PARAMETERS:
0.50000D+01   0.50000D-01   0.72113D-01   0.50000D-01
```

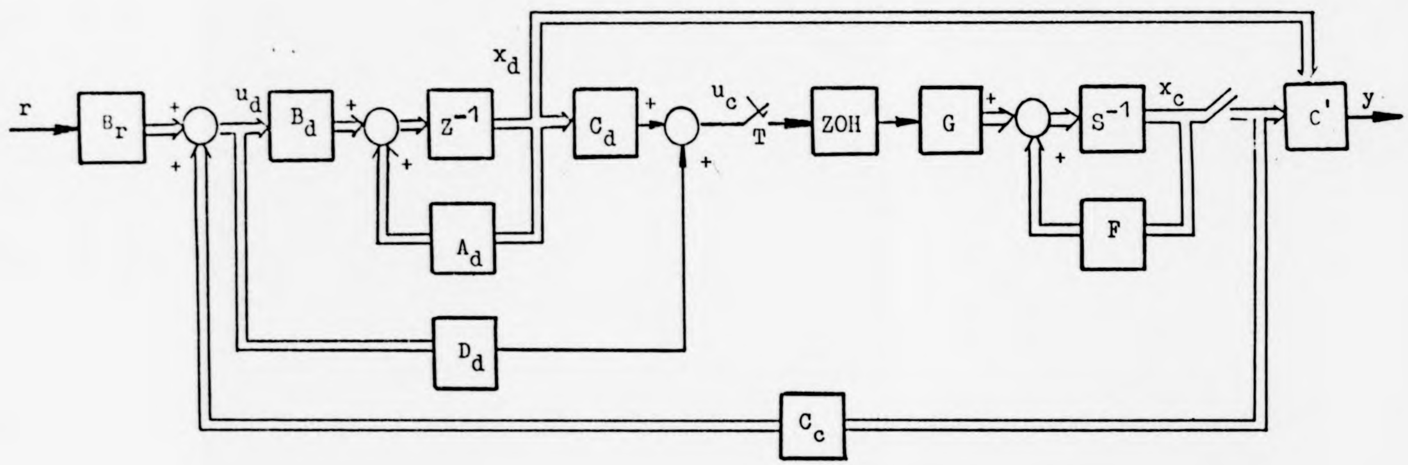

Fig. 7—1 Digital control system

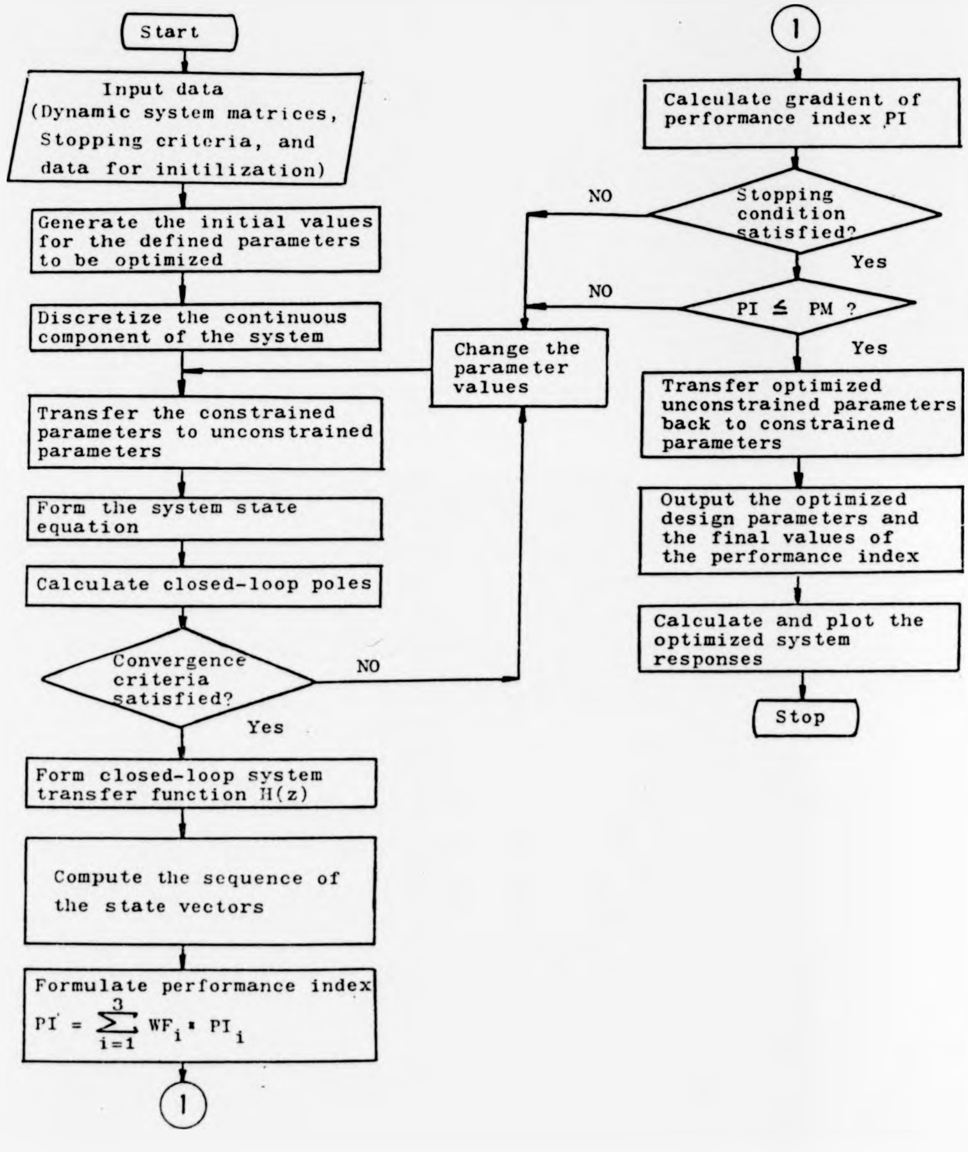

Fig. 7-2 Computer program flow chart

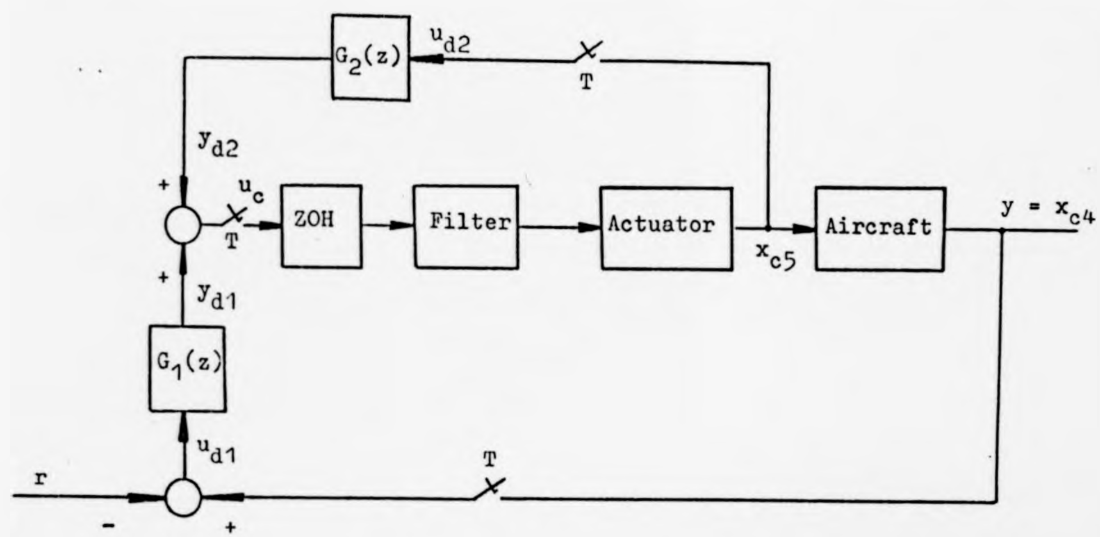

Fig. 7—3 Space shuttle pitch axis digital control system

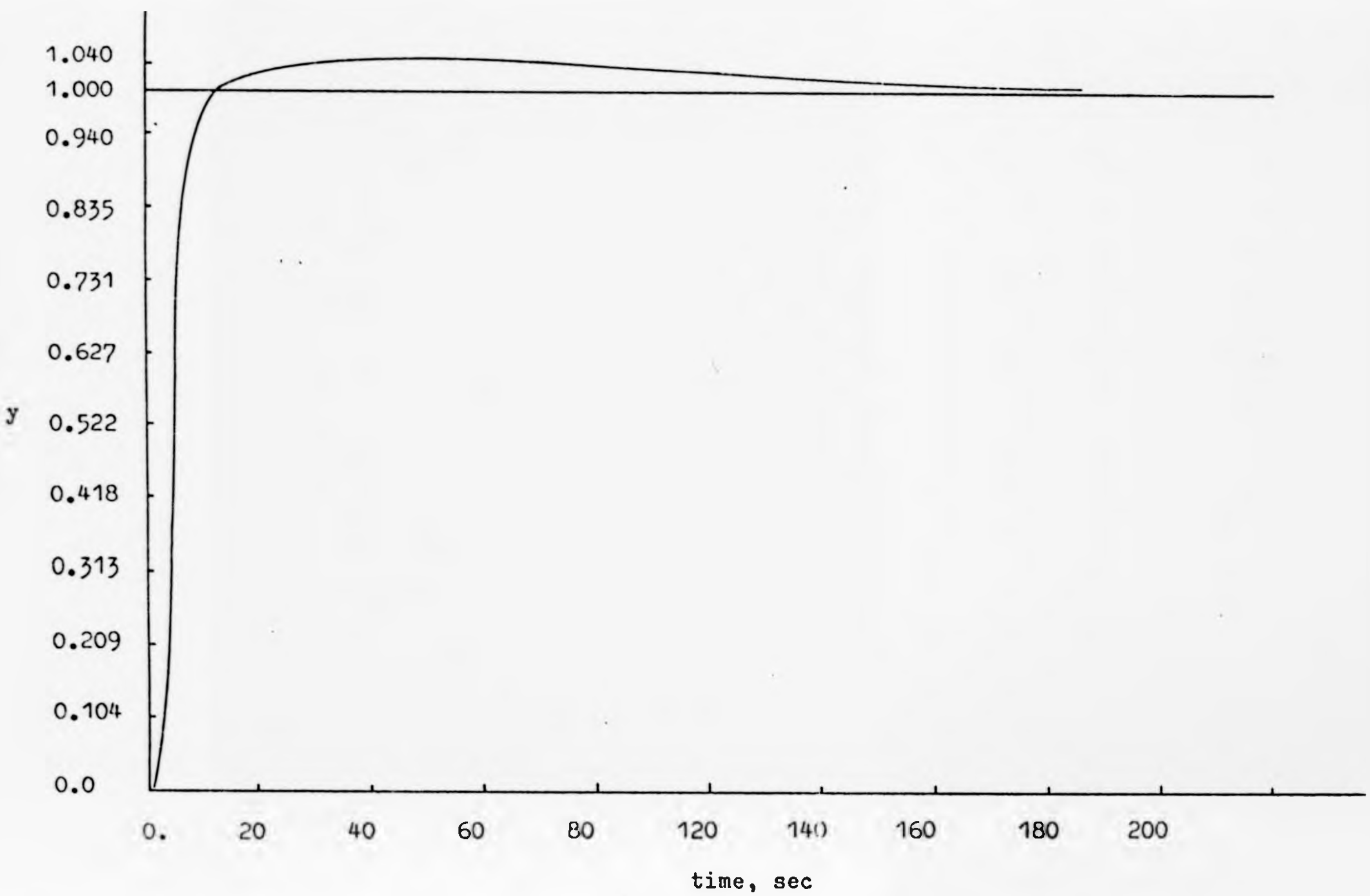

Fig. 7—4 Unit step response of design example

7-5 Concluding remarks

This chapter covered a CAD technique for digital control systems with finite settling time. A time-domain performance index PI is introduced to penalize the system error, the actuating force, and the closed-loop poles. The Quasi-Newton nonlinear programming algorithm is applied to have the parameter optimized design. A CAD program has been implemented in FORTRAN.

Chapter – 8 CONCLUDING COMMENT

This thesis has presented some new results in two key areas of control theory and control engineering : identification and design. On identification (Part–I, Chapters 1 – 5), the main contribution of the thesis has been to establish a mathematical framework so that linear discrete and continuous systems as well as time-varying and non-linear systems can be identified through the same algorithm.

The method presented in Chapter 2 is applicable in the estimation of unknown parameters of multiple-input multiple-output linear time-invariant continuous systems and a class of time varying systems providing the system order is known. All system states are required to be accessible for measurement. The method is based on using a set of integrators and a few relatively simple matrix operations. The integration of state equation and output equation has the advantage of decreasing the effect of measurement noise. In this method the computations required are one single matrix inversion and one matrix multiplication. An identification algorithm and a computer program have been developed in Chapter 2, and four numerical examples considered. These examples, overdamped, underdamped, undamped and unstable systems, have been included to demonstrate the general applicabilty of the proposed method to a wide variety of systems. The general conclusion derived from these examples was that the estimation error could be kept within acceptable error limits providing one condition was met in the choice of sampling inteval (T). The condition imposed on T was that the required number of sampled input-output data must cover the entire range of the transient response of the system. Some typical storage requirement and processing times for this algorithm are given below :

System Order	CPU time , Secs	Memory usage , Kbyte
2	1.36	16
3	1.4	21
4	1.46	30
5	1.51	45

In this method no special test signal is required, which makes the method applicable for on-line estimation as well as for the identification to identify any parameter variation which may occur during the normal operation of the system under consideration.

The method that we have developed in Chapter 3 is more attractive, in that it can identify the order as well as the parameters of multiple-input multiple-output systems from input-output observations, the only computations required is a single matrix inversion and a single matrix multiplication. This method again is based on the multiple integration of dynamical equation; this integration filters out the some of effects of system noise on identification.

An identification algorithm and a computer program have been developed using the theory formulated in Chapter 3. A close observation of Figs. 3-3, 3-4, 3-5 and 3-6 related to the several types of systems indicates that the agreement between the actual system parameters and estimated values is very close. In fact, the scalar error quantities S_A, S_B, S_O and S are below 0.001 for all above examples over a wide range of sampling intervals. An important fact which comes out of Figs. 3-3 $\sim$ 3-6, is that it permits the collection of input-output data at a much faster sampling rate. This has considerable practical significance as the proposed algorithm eliminates the need for a long period of time for data collection. In summary, it may be stated that the algorithm developed here is acceptable for application where a low-cost mechanism is needed for fast identification of system parameters from on-line data. The other advantage of the method is that the engineer has the flexibility of choosing the sampling interval from a wide range of values. Figs. 3-3 $\sim$ 3-6 demonstrate the validity of this statement for overdamped-underdamped-undamped as well as unstable systems. This makes the method of Chapter 3 more attractive than other well established estimation techniques.

Some typical storage requirement and processing times for this algorithm are given below

system order	CPU time (seconds)	memory usage (kbyte)
2	2.05	193
3	2.15	210

This method uses only the system inputs and outputs, and only biased sinusoidal test signal. This makes this method (like the one in Chapter 2) suitable for the on-line estimation of parameter variation which may occur during the normal operation of the system under consideration.

In the method derived in Chapter 3, no prior information about the initial conditions is required. Furthermore, in addition to the parameter determination, the initial conditions can be determined by this scheme. The drawback of this method is that we have to construct a fairly high order matrix $V(T)$ of dimension $n(m+2)\times n(m+2)$.

In practice, even when it is possible to identify the parameters of complex and high order systems, the analysis, optimization and adaptation would require a large amount of computation. One way of overcoming these computational difficulities is to use a low order model of the high order system which is computationally and analytically more tractable than the actual system, yet still provides sufficient information about the original system.

Some related results on reduction of high order system have been considend in Appendix 4. The method proposed in Appendix 4 may be used to obtain a low order model for a given high order system from experimental data, using only the system inputs and outputs. A close observation of Tables A4 – 1 and A4 – 2 related to the examples of reduction of high order system indicate that the proposed method produces reduced order models which are good approximations of the actual system over a wide range of sampling intervals.

The extended method that we have developed in Chapter 4 is applied to identify some special forms of non-linear and time varying systems. Two special cases are considered here, (a) When the matrices A(t), B(t) and C(t) are known to be in certain special forms (time-varying systems, the Euler type). If the time-varying elements of the system to be identified are known to be slowly varying, i.e., the parameters remain constant over the time interval $[t_0, t_f]$, then such a system can be considered as time invariant and both methods of Chapters 2 and 3 can be applied at successive intervals of time by resetting the integrators at the end of each interval. (b) Non-linear systems considered here are assumed to be in the form of Duffing's equation. With some modifications, the method of Chapter 3 has been applied to identify the unknown parameters of such systems.

The method that we have developed in Chapter 5 is applicable to discrete-time systems, the order determination is based on the rank difference between two appropriately constructed matrices.

In this method of linear discrete-time systems, the only computations required is a single matrix inversion. The order determined may be incorrect when unsuitable input sequences were used. In fact, since sampling length N is finite, there is always a possibility that the actual order of the system is higher than the one estimated. The basic procedure employed for the identification of unknown parameters is based on a special output data vector $\underline{y}(K;N)$ — a linear combination of the inputs-outputs data.

Since the computational procedure presented in Chapters 2 — 5 is a one-shot procedure, where no iterations are required, the time required for identification is considerably less than the iterative methods. This is a distinct advantage of the methods proposed in this thesis over established iterative techniques.

The method developed in Chapter 6 is applicable to the design of a microprocessor-based output deadbeat controller for digital servo systems with finite settling time. Based on custom-designed hardware, this chapter describes the implementation of a digital lead-lag compensator and a digital PID controller.

The method that we have developed in Chapter 7 is intended for the design of digital control systems with finite settling time. A time-domain performance index(PI)is introduced to penalize the system error, the actuating force, and the closed-loop poles. The Quasi-Newton nonlinear programming algorithm is applied to obtain the parameter optimized design. A CAD program has been implemented in FORTRAN.

It is worth indicating that this thesis has been concerned with systems where the amount of noise is insignificant and thus the measurement data for identification or control need not be filtered. In view of the relative success of the identification techniques and design methods presented here, it is hoped that future work would continue with a view to extending these results to more realistic situations. Some factors which should be included in extending these results are: noisy data, severe parameter variations and the methods of design with sensor failures.

REFERENCES

[1] D. D. Donalson and C.T. Leondes, " A model-reference Parameter Tracking Technique for Adaptive Control systems," AIEE Trans. Pt. 2, PP.241 — 262, September, 1963.

[2] K. S. Prasanna Kumar and Sridhar " On the identificafion of control systems by the quasilinearization method," IEEE Trans., Vol. AC—9, No. 2, April, 1964, PP. 151—154.

[3] N. N. Puri and C. N. Waygant " Transfer function tracking of a linear time-varying system by means of auxiliary simple lag network," IEEE Trans. Pt. 2, January,1964, PP. 170 — 172.

[4] A.E. Rosenberg and D. W. C. Shen, " Regression analysis and its application to the system identification problem," Proc. Joint Automatic Control Conference, Minn. June, 1963, PP. 446 — 451.

[5] B. L. Ho and R. E. Kalman, " Effective construction of linear state variable models from input outnut data," Proc. Third Ann. Allerton Conference on circuit and System theory, 1965, PP. 449 — 459.

[6] B. Gopinath " On the identification of linear time-invariant systems from input-output data " Bell Syst.

Tech. J., Vol. 48, May-June 1969.

[7] K. J. Khatwani and J. S. Bujwa, " Identification of linear time invariant systems using exponential signals," " IEEE Trans. Automatic Control, Vol. AC—20, February, 1975.

[8] K. S. Narendra and S. S. Tripathi, " Identification and optimization of aircraft dynamics," J. Airer., Vol. 10, PP.193 — 199, April, 1973.

[9] G. Luders and K. S. Narendra, " Stable adaptive schemes for state estimation and identification of linear systems," IEEE Trans. Automatic Control, Vol. AC—19, No. 6, December, 1974.

[10] T. C. Hsia and V. Vimolvanich, " An on-line technique for system identification," " IEEE Trans. Automatic Control, Vol. AC, PP. 92 — 96, February, 1969.

[11] A. Sherif and M. Y. Wu, " Identification of linear dynamical systems," Int. J. Control, 1974, Vol. 19, No.1, PP. 185 — 192.

[12] H. Aded : " Optimal control and identification of distillation town," ph. D Dissertation, University of Colorado, Boulder, Colorado. 1976.

[13] D. Graupe, D. J, Krause and J. B. Moore, " Identification of Auto-regressive Moving-Average Parameters of Time Series", IEEE Trans. Auto. Control, Vol. AC—20, PP.104 — 107, Feb. 1975.

[14] K. J. Astrom and P. Eykhoff, " System Identification ··· A Survey", Automatica, Vol. 7, pp. 123 — 162, Mar.1971.

[15] J. C. Chow, " On the Estimation of the order of a Moving-Average Process". TEEE Trans. Auto. Control, Vol. AC—17, pp.386 — 387, June 1972.

[16] J. C. Chow, "On Estimating the Orders of an Autoregressive Moving-Average Process with Uncertain Observations", IEEE Trans. Auto. Control, Vol. AC—17, pp.707 — 709, Oct. 1972.

[17] C. M. Woodside, " Estimation of the Order of Linear Systems". Automatica, Vol.7, pp. 727 — 733, Nov. 1971.

[18] H. Unbehauen and B. Gohring, " Tests for Determining Model Order in Parameter Estimation", Automatica, Vol.10, pp.233 — 244, May 1974.

[19] A. J. W. Van Den Boom and A. W. M. Van Den Enden, "The Determination of the Orders of Process-and Noise Dynamics", Automatica, Vol. 10, pp.245 — 256, May 1974.

[20] R. K. Mehra, "On-Line Identification of Linear Dynamics Systems with Applications to Kalman Filtering", IEEE Trans. Auto. Control, Vol. AC—16, pp.12 — 21, Feb.1971.

[21] E. Tse and H.L. Weinert, " Structure Determination and Parameter Identification for Multi-Variable stochastic Linear Systems", IEEE Traus. Auto. Control, Vol. AC—20, pp.603 — 613, Oct.1975.

[22] V. M. Popov, " Invariant Description of linear Time

Invariant Controllable Systems", SIAM J. Control, Vol.10, pp.252—264, May 1972.

[23] M. J. Denham, "Canonical Forms for the Identification of Multivariable Linear Systems", IEEE Trans. Auto. Control, Vol. AC—19, pp.646—656, Dec. 1974.

[24] R. Guidorzi, "Canonical Structure in the Identification of Multivariable Systems", Automatica, Vol.11, pp.361—374, July 1975.

[25] R. K. Mehra, "On the Identification of variances and Adaptive Kalman Filtering", IEEE Trans. Auto.Control, Vol. AC—15, pp.175—184, Apr.1970.

[26] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Trans. ASME, J. Basic Engrg., Ser. D, Vol. 82, pp. 35—45, Mar.1960.

[27] T. P. McGarty, "Stochastic Systems and state Estimation", John Wiley & Sons, Inc., New York, N.Y.1974.

[28] J. L. Melsa and A. P. Sage, "An Introduction to Probability and Stochastic Processes", Prentice-Hall, Inc., Englewood Cliff's, New Jersey, 1973.

[29] T. Kailath, "An Innovations Approach to least-squares Estimation……Part 1 : Linear Filtering in Additive white Noise", IEEE Trans, Auto. Control, Vol.AC—13, pp.646—654, Dec. 1968.

[30] T. Kailath, "The Innovations Approach to Detection and Estimation Theory", Proc. IEEE, Vol. 58, No.5,

pp.680 — 695, May, 1970.

[31] S. J. Merhav and E. Gabay, "On Simultaneous structure and Parameter Identification of Linear Dynamical Systems", IEEE Trans. Auto. Control, Vol. AC—19, pp.401 — 404 , Aug. 1974.

[32] S. J. Merhav and E. Gabay, "Unbiased Parameter Estimation by Means of Autocorrelation Functions", IEEE Trans. Auto. Control, Vol. AC—20, pp.368 —372, June 1975.

[33] J. M. Mendel, "Multistage Least-Square Parameter Estimators", IEEE Trans. Auto. Control, Vol. AC—20, pp.775 — 782 , Dec.1975.

[34] J. H. Anderson, 1967, Proc. Instn. Elect. Engrs. (London), 114, 1014.

[35] C. F. Chen and L. S. Shieh, 1968 Int. J. Control, Vol.8, No. 6, 561 —570.

[36] E. J. Davison, 1966 IEEE Trans. Auto. Control 11, 93, 1968, Ibid., 13, 215.

[37] W. R. Evans, 1954, Control Systems Dynamics (New York, McGraw-Hil' Book Co.)

[38] N. K. Sinha and W. Pille, Int. J. Control, 1971, Vol.14, No.1, 111 —118.

[39] K. Chen, 1957 Trans. Am. Inst. elec. Engrs. 76, 80.

[40] G. A. Biernson, 1956, Trans. Am. Inst. elect. Engrs., 75, 253.

[41] M. R. Chidambara, 1969, Proc. Joint Automatic Control

Conference, p.669.

〔42〕 N. K. Sinha and G.T. Bereznai, Int. J. Control, 1971, Vol.14, No. 5, 951 —959.

〔43〕 R. Hooke and T. A. Jeeves, 1961, J.A.C.M., 8, 212.

〔44〕 T. Yahagi, IEEE Trans. Circuit and systems, Vol. CAS—24, No. 10, October, 1977.

〔45〕 M.DE LA SEN, 1983, Int. J. Systems SCI. Vol.14, No. 4, 367 —383.

〔46〕 DELA SEN, M., 1982 2, Electron. Lett., 18, 311; 1982 b, Proc. of the IEEE Symposium on Circuits and Systems, Vol. 2 (New York: IEEE 82CH1681—b), P.393.

〔47〕 DELA SEN, M., and DORMIDO, S., 1979, Proceedings of the Seventeenth Annual Allerton Conference, edited by J. B. Cruz, Jr. and F. F. Preparata (Urbana-champaign: Department of Electrical Engineering and Coordinated Science Laboratory of the University of Illinois), p.206; 1981 2, Preprints of the Eighth IFAC World Congress, Vol. VI, 71; 1981 b, Electron. Lett., 17, 922.

〔48〕 DORMIDO, S., and DE LA SEN, M., 1979, IEEE Trans. autom. Control, 24, 634; 1982, Progress in Cybernetics and Systems Methodology, edited by R. Trappl, G. J. Klir and F.R. Pichler (Washington: Hemisphere Publishing Corporation), p.367.

〔49〕 DORMIDO, S., DE LA SEN, M., and MELLADO M., 1978 a, Revista de Information y Automatica, 38, 13:

1978 b, Ibid., 38, 22 (in Spanish); 1980, Advances in Control, edited by D.G. Lainiotis and N.S. Tzannes (Dordrecht : D. Reidel Publishing Company), p.37.

[50] EYKHOFF, P., 1971, System Parameter and State Estimation (London : Wiley).

[51] FISHER, F.M., 1966, The Identification Problem in Econometrics (New York : McGraw-Hill).

[52] GREWAL, M.S., and GLOVER, K., 1976, IEEE Trans. autom. Control, 21, 833.

[53] GLOVER, K., and WILLEMS, J.C., 1974, IEEE Trans. autom. Control, 19, 640.

[54] HSIA, T. C., 1972, IEEE Trans. autom. Control, 17, 830; 1974, Ibid 19, 39.

[55] ORTEGA, J. M., 1972, Numerical Analysis (New York: Academic Press).

[56] PARASKEVOPOULOS, P. N., 1979, IEEE Trans. Ind, Electron. Control Instrum., 26, 234; 1980 Ibid., 27,242.

[57] PAZ, M.B., DE LA SEN, M., DORMIDO, S., and MELLADO, M., 1982, Electron. Lett., 18, 404.

[58] REID, J. C., 1977, IEEE Trans. autom. Control, 22, 242.

[59] REID, J. C., MAYBECK, P. J., ASHER, P. B., and DILLOW, J. D., 1976, J. Franklin Inst., 301,123.

[60] ROTHENBERG, T. J., 1971, Econometrica, 39, 577.

[61] THOWSEN, A., 1978, Int. J. Systems Sci., 9, 813.

[62] VAJDA, S., 1979, IEEE Trans. autom. Control, 24, 495.

[63] ZADEH, L. A., and DESOER, C.A., 1963, Linear System Theory : The state Space Approach (New York : McGraw-Hill).

[64] B. D. O. Anderson, "Adaptive Identification of Multiple-input Multiple-output plants." Proceedings, 1974 IEEE Conference on decision and control, pp.273—281, Phoenix, November 20—22.

[66] K. J. Astrom and P. Eykhoff, " System Identification A Survey." Automatica, Vol. 7, pp.123—162, 1971.

[67] M. L. Bakker, Determination of the state variable A-Matrix by direct measurement. IEEE Trans. Automatic Control, July, 1966, p.610.

[68] R. L. Carrol and R. V. Monopoli, "Model reference adaptive control estimation and identification using only input and output signals," Proc. of IFAC/75 Conf., August, 1975.

[69] C. T. Chen, Introduction of linear system theory Holt, Rhinehart, and Winston, New York, 1978.

[70] D. Denery, "An identification algorithm that is insensitive to initial parameter estimates," AIAA J., Vol.9, pp. 371 — 377, 1971.

[71] R. C. Dorf. Modern Control Systems, Addison-Wesley, 1967.

[72] D. K. Faddeu and U. N. Faddeva, Computational Methods of linear algebra. W. H. Freeman and Company, 1963.

[73] F. R. Gantmakher, The Theory of Matrices, Chelsea, 1959.

[74] I. Gustavsson, "Comparison of different methods for identification of industrial proccess." Automatica, Vol. 8, pp. 127—142, 1972.

[75] Y. C. Ho, "An approach to the identification and control of linear dynamic systems with unknown parameters." IEEE Trans. Automatic Control, 1963., pp.255—256

[76] T. C. Hsia, "On Sampled Data Approach to Parameter Identification of Continuous Linear Systems, " IEEE Trans. Automatic Control, Vol. 7, April, 1972.

[77] B. C. Kuo, Discrete-Data Control Systems, Prentice-Hall, Inc., 1970.

[78] R. E. Kalman, "Mathematical description of linear dynamical systems," SIAM J. Control, Vol. 1, 1963, pp. 152 — 192.

[79] G. G. Lendaris, "The identification of linear systems," AIEE Trans. pt. 2, pp. 231—242, Sept. 1962.

[80] K. Ogata, Modern Control Engineering, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1970.

[81] K. G. Oza and E. I. Jury, "Adaptive algorithm for identification problem," Automatics, Vol.6, pp.795—799, November, 1970.

[82] P. R. Rao, " Identification of linear non-stationary dynamical systems," INT. J. Systems Sci., 1974, Vol. 5, No. 2, pp.117 — 129.

[83] G. Strim, G. L. Hartmann and R. C. Hendrick, " Adaptive Control laws for F—8 Flight test," IEEE Trans. Aut. Cont., Vol. AC—22, No. 5, October, 1977.

[84] P. Stoica and T. Soderstrom, " A method for the identification of linear systems using the generalized least squares principle." IEEE Trans. Aut Cont., Vol.AC—22, No. 4, August, 1977.

[85] W. A. Wolovich, "The determination of state space representations for linear multivariable systems." Automatica, Vol. 9, pp. 47 — 106, 1973.

[86] L. A. Zadeh and C. A. Desou, Linear system theory, McGraw-Hill, New York, 1963.

[87] Z. V. Rekasius and F. M. Brash, "A Theoretical Look at Modeling and Identification", Northwestern University, Evanston, Illinois, 1977.

[88] B. L. Ho and R. E. Kalman, "Effective Construction of Linear State-variable Models from Input/output Functions", Regelungstechnik, Vol. 14, pp. 545 — 548, 1966.

[89] R. E. Kalman, P. L. Falb and M. A. Arbib, "Topics in Mathematical System Theory", Chapter 10, McGraw-Hill Book Company, New York, 1969.

[90] A. J. Tether, " Construction of Minimal Linear State-

Variable Models from Finite Input/Output Data", IEEE Trans. Auto. Control, Vol. AC—15, pp.427—436, Aug. 1970.

[91] M. A. Budin, "Minimal Realization of Discrete Linear Systems from Input/Output Observations", IEEE Trans. Auto. Control, Vol. AC—16, pp. 395—401, Oct. 1971.

[92] E. Emre, L. M. Silverman and K. Glover, "Generalized Dynamic Covers for Linear System with Applications to Deterministic Identification and Realization Problems", IEEE Trans. Auto. Control, Vol. AC—22, pp.26—35, Feb.1977.

[93] B. Gopinath, "On the Identification of Linear Time-Invariant Systems from Input-Output Data", The Bell System Technical Journal, Vol. 48, No. 5, pp.1101—1113, May-June 1969.

[94] A. Papoulis, "Probability, Random Variables, and Stochastic Processes", McGraw-Hill Book Company, New York, 1965.

[95] T. T. Soong, "Random Differential Equations in Sciences and Engineering", Academic Press, Inc., New York, N. Y. 1973.

[96] C. R. Rao and S. K. Mitra, "Generalized Inverse of Matrices and its Applications", John Wiley & Sons, Inc., New York, N. Y. 1971.

[97] H. R. Schwartz, H. Rutishauser and E. Stiefel, Translated

by P. Hertelendy, "Numerical Analysis of Symmetric Matrices", Prentice-Hall, Inc., Englewood Cliff's, New Jersey, 1973.

[98] N. S. Hsu and B. Cheng, "Identification of non-linear distributed system via block-pulse functions," Int. J. Control, Vol. 36, No. 2, 281—291, 1982.

[99] Y. C. Wu and Z. V. Rekasius, IEEE Trans. Automatic Control, Vol. AC—25, No. 3, June. 1980.

[100] R. S. Roth, "Techniques in the identification of deterministic systems," IEEE Trans. Automatic Control, Vol. AC—26, No. 5, Oct., 1981.

[101] R. Sivan: IEEE Trans. Automatic Control AC—10,(1965), 193

[102] S.P. Bhattacharrya, et.: Information and Control 20, (1972), 135

[103] L.M. Silverman, et.: SIAM J. Control 9,(1971), 199

[104] G. Guardabassi, et.: Internat. J. Control 15, (1972), 758

[105] P. L. Falb, et.: IEEE Trans. Automatic Control AC-12, (1967), 651

[106] P.K. Sinha,: IEEE Trans. Automatic Control AC—24 (1979), 476

[107] J.W. Howze,: IEEE Trans. Automatic Control AC—18 (1973), 44

〔108〕T.L. Duffield, et.:IEEE Trans. Automatic Control AC—22 (1977), 142

〔109〕W. M. Wonham,: IEEE Trans. Automatic Control AC—12 (1967), 660

〔110〕J.B. Peason,etc.: IEEE Trans. Automatic Control AC—15 (1970), 34

〔111〕W.A. Wolovich,: IEEE Trans. Automatic Control AC—20, (1975), 148

〔112〕W. A. Wolovich,: IEEE Trans. Automatic Control AC—18, (1973), 544

〔113〕J.J. D'azzo and C.H. Houpis, Linear Control System Analysis and Design. New York: McGraw-Hill, 1981

〔114〕J. A. Cazow and H.R. Martens, Discrete-Time and Computer Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1970

〔115〕C.L. Phillips and H.T. Nagle, Jr., Digital Control System Analysis and Design. Englewood Cliffs, NJ: Prentice-Hall, 1984

〔116〕R. Fetcher, practical Methods of Optimization. New York: John Wiley & Sons, 1980

〔117〕E. I. Jury, Sample-Data Control Systems. New York: John Wiley & Sons, 1958

〔118〕F. -Y. Shih, " Design algorithms for digital control systems with deadbeat unit step response, " IEE Proc., vol. 130, Pt. D, no. 3, pp. 119-127, May 1983

〔 119 〕K. J. Astrum and B. Wittenmark, Computer-Controlled Systems. Englewood Cliffs, NJ :Prentice-Hall, 1984.

〔 120 〕R. Isermann, Digital Control Systems. Berlin: Springer-Verlag, 1981.

〔 121 〕P. B. Deshpande and R. H. Ash, Elements of Computer Process Control with Advanced Control Applications Research Triangle Park, NC: Instrument Society of America, 1981.

〔 122 〕J. A. Cabzow and H. R. Martens, Discrete-Time and Computer Control Systems Englewood Cliffs, NJ : Prentice-Hall, 1970.

〔 123 〕C.L. Phillips and H. T. Nagle, Jr., Digital Control System Analysis and Design. Englewood Cliffs, NJ : Prentice-Hall, 1984.

〔 124 〕N. J. Krikelis and S. D. Fassois, " Microprocessor implementation of PID controllers and Lead-Lag compensators," IEEE Trans. Ind. Electron. vol. IE-31, no. 1, pp. 79-85, Feb. 1984.

〔 125 〕F. Y: Shih, " Design algorithms for digital control systems with deadbeat unit step response," IEE Proc., vol 130, Pt. D. no 3, pp. 119-127, May 1983.

APPENDIX 1

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                          PROGRAM A                       C
C   THIS PROGRAM IS USED TO IDENTIFY MIMO KNOWN            C
C   ORDER SYSTEMS FROM STATES.                             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT REAL*8(A-H,O-Z)
      EXTERNAL U1,U2
      INTEGER P
      DIMENSION A(10,10),B(10,10),C(10,10),V(10),U(10),F(10)
      DIMENSION YA(10,11),UU(10,10)
      DIMENSION PA(10,10),VO(10),VT(20,20),ET(20,20),X1(10),G(20,20)
      DIMENSION AZ(10,10),BV(10,10),TB(10,10)
601   FORMAT(1H1)
701   FORMAT(10X,' SYSTEM ORDER = ',I3)
23    FORMAT(/20X,' ****  INITIAL VALUE : ',//)
17    FORMAT(//20X,' ****  MATRIX A IS AS FOLLOWS :',/)
18    FORMAT(//20X,' ****  MATRIX B IS AS FOLLOWS :',/)
19    FORMAT(//20X,' ****  MATRIX C IS AS FOLLOWS :',/)
702   FORMAT(10X,' SAMPLING PERIOD = ',F8.5)
16    FORMAT(10X,6E12.4)
10    FORMAT(3I2,F8.5)
15    FORMAT(8F9.4)
C     H : TIME INTERVAL
1     READ(5,10,END=9999)N,M,P,H
      READ(5,15)(V(I),I=1,N)
      DO 94 I=1,N
94    VO(I)=V(I)
      WRITE(6,601)
      WRITE(6,701)N
      WRITE(6,702)H
      WRITE(6,23)
      WRITE(6,16)(V(I),I=1,N)
      WRITE(6,17)
      DO 20 I=1,N
      READ(5,15)(A(I,J),J=1,N)
20    WRITE(6,16)(A(I,J),J=1,N)
      WRITE(6,18)
      DO 30 I=1,N
      READ(5,15)(B(I,J),J=1,M)
30    WRITE(6,16)(B(I,J),J=1,M)
      WRITE(6,19)
      DO 40 I=1,P
      READ(5,15)(C(I,J),J=1,N)
40    WRITE(6,16)(C(I,J),J=1,N)
      CALL STMST(N,H,A,PA)
      WRITE(6,703)
703   FORMAT(1H1,20X,' *** COMPUTED PHI MATRIX *** '/)
      DO 704 I=1,N
704   WRITE(6,16)(PA(I,J),J=1,N)
      CALL QT(N,M,A,PA,B,TB)
      WRITE(6,705)
705   FORMAT(//20X,' *** COMPUTED THETA MATRIX *** '/)
      DO 706 I=1,N
```

```
706   WRITE(6,16)(TB(I,J),J=1,N)
      DO 134 K=1,M+N
      HK=H*(K-1)
      CALL DX(N,M,K,PA,TB,H,V,F)
      DO 25 I=1,N
      VT(I,K)=F(I)
25    V(I)=F(I)
134   CONTINUE
      WRITE(6,707)
707   FORMAT(//20X,' *** COMPUTED STATE MATRIX ***'/)
      DO 708 I=1,N
708   WRITE(6,16)(VT(I,J),J=1,N+M)
      EPS=0.000001
      ZE=0.
      DO 42 I=1,N+M
      TT=I*H
      CALL SIMPSN(ZE,TT,EPS,U1,VI)
      CALL SIMPSN(ZE,TT,EPS,U2,VII)
      UU(1,I)=VI
      UU(2,I)=VII
      ET(1+N,I)=VI
42    ET(2+N,I)=VII
      WRITE(6,709)
709   FORMAT(//20X,' *** COMPUTED R(T) MATRIX ***'/)
      DO 710 I=1,M
710   WRITE(6,16)(UU(I,J),J=1,N+M)
      DO 43 I=1,N+M
      DO 43 J=1,N
43    VT(J,I)=VT(J,I)-V0(J)
      DO 44 I=1,N
      DO 44 J=1,N+M
      BV(I,J)=0.
      DO 44 K=1,M
44    BV(I,J)=BV(I,J)+B(I,K)*UU(K,J)
      DO 45 I=1,N
      DO 45 J=1,N+M
45    AZ(I,J)=VT(I,J)-BV(I,J)
      EPS=1.0E-19
      INDIC=-1
      DETER=SIMUL(N,A,X1,EPS,INDIC,10)
      DO 47 I=1,N
      DO 47 J=1,N+M
      ET(I,J)=0.
      DO 47 K=1,N
47    ET(I,J)=ET(I,J)+A(I,K)*AZ(K,J)
      WRITE(6,715)
715   FORMAT(//20X,' *** COMPUTED E(T) MATRIX ***'/)
      DO 716 I=1,N
716   WRITE(6,16)(ET(I,J),J=1,N+M)
      DO 54 I=N+1,N+P
      DO 54 J=1,N+M
      VT(I,J)=0.
      DO 54 K=1,N
54    VT(I,J)=VT(I,J)+C(I-N,K)*ET(K,J)
      WRITE(6,711)
711   FORMAT(//20X,' *** COMPUTED F(T) MATRIX ***'/)
```

```
      DO 712 I=N+1,N+P
712   WRITE(6,16)(VT(I,J),J=1,N+M)
      WRITE(6,3330)
3330  FORMAT(1H1,//20X,' *** COMPUTED E(T)  MATRIX ***'/)
      DO 3716 I=1,N
3716  WRITE(6,16)(ET(I,J),J=1,N+M)
      DO 3710 I=1,M
3710  WRITE(6,16)(JJ(I,J),J=1,N+M)
      WRITE(6,3707)
3707   FORMAT(//20X,' *** COMPUTED V(T) MATRIX ***'/)
      DO 3708 I=1,N+P
3708   WRITE(6,16)(VT(I,J),J=1,N+M)
      NPM=N+M
      DETER=SIMUL(NPM,ET,X1,EPS,INDIC,20)
      IF (DETER.NE.0.)GO TO 98
      WRITE(6,73)
73    FORMAT(10X,'$$$$$,NO INVERSE ')
      GO TO 1
98    WRITE(6,600)
600   FORMAT(1H1,10X,' **** FINAL RESULTS G ****',//)
      DO 222 I=1,N+P
      DO 222 J=1,M+N
      G(I,J)=0.
      DO 222 KI=1,M+N
222   G(I,J)=G(I,J)+VT(I,KI)*ET(KI,J)
      DO 333 I=1,N+M
333   WRITE(6,16)(G(I,J),J=1,N+M)
717   FORMAT(//10X,' *** ESTIMATED A MATRIX ***',//)
718   FORMAT(//10X,' *** ESTIMATED B MATRIX ***',//)
719   FORMAT(//10X,' *** ESTIMATED C MATRIX ***',//)
720   FORMAT(//10X,' *** ESTIMATED D MATRIX ***',//)
      WRITE(6,717)
      DO 721 I=1,N
721   WRITE(6,16)(G(I,J),J=1,N)
      WRITE(6,718)
      DO 722 I=1,N
722   WRITE(6,16)(G(I,J),J=N+1,N+M)
      WRITE(6,719)
      DO 723 I=N+1,N+P
723   WRITE(6,16)(G(I,J),J=1,N)
      WRITE(6,720)
      DO 724 I=N+1,N+P
724   WRITE(6,16)(G(I,J),J=N+1,N+P)
      GO TO 1
9999  STOP
      END
```

```
      SUBROUTINE SIMPSN(A,B,EPS,F,TSUM)
      IMPLICIT REAL*8(A-H,O-Z)
      H=B-A
      SUM1=F(A)+F(B)
      TSUM1=4*H*SUM1/3.
      M=1
    1 H=H/2.
      M=M*2
      SUM2=0.
      J=1
    4 SUM2=SUM2+2*F(A+J*H)
      IF (J+1-M) 2,3,3
    2 J=J+2
      GO TO 4
    3 SUM21=SUM1+SUM2*2
      SUM1=SUM1+SUM2
      TSUM=SUM21*4*H/3.
      IF (DABS(TSUM-TSUM1)-EPS)5,6,6
    6 TSUM1=TSUM
      GO TO 1
    5 RETURN
      END

      FUNCTION U1(X)
      REAL*8 U1,X
      U1=1.+DSIN(1.2*X)
      RETURN
      END

      FUNCTION U2(X)
      REAL*8 U2,X
      U2=2.+DCOS(2.4*X)
      RETURN
      END
```

```
      SUBROUTINE QT(N,M,A,PA,B,TB)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AA(10,10),A(10,10),PA(10,10),B(10,10),P(10,10)
      DIMENSION X(10),PB(10,10),TB(10,10)
      EPS=1.0D-19
      INDIC=-1
      DO 5 I=1,N
      DO 5 J=1,N
5     AA(I,J)=A(I,J)
      DET=SIMUL(N,AA,X,EPS,INDIC,10)
      DO 10 I=1,N
      DO 10 J=1,N
      P(I,J)=PA(I,J)
      IF (I.EQ.J) P(I,J)=P(I,J)-1.
10    CONTINUE
      DO 20 I=1,N
      DO 20 J=1,N
      PB(I,J)=0.
      DO 20 K=1,N
20    PB(I,J)=PB(I,J)+AA(I,K)*P(K,J)
      DO 30 I=1,N
      DO 30 J=1,N
      TB(I,J)=0.
      DO 30 K=1,N
30    TB(I,J)=TB(I,J)+PB(I,K)*B(K,J)
      RETURN
      END


      SUBROUTINE STMST(N,H,A,PH)
C     REAL*8 A,PH,AT,SAT,AC,H,R
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(10,10),PH(10,10),AT(10,10),SAT(10,10),AC(10)
      DO 43 I=1,N
      DO 43 J=1,N
43    PH(I,J)=0.
      DO 44 I=1,N
44    PH(I,I)=1.
      DO 46 I=1,N
      DO 46 J=1,N
      AT(I,J)=A(I,J)*H
46    SAT(I,J)=AT(I,J)
      R=2.
      DO 47 NK=1,30
      DO 48 I=1,N
      DO 48 J=1,N
48    PH(I,J)=PH(I,J)+SAT(I,J)
      DO 49 I=1,N
      DO 50 J=1,N
      AC(J)=0.
      DO 50 K=1,N
50    AC(J)=AC(J)+SAT(I,K)*AT(K,J)
      DO 49 J=1,N
49    SAT(I,J)=AC(J)/R
47    R=R+1.
      RETURN
      END
```

```
      SUBROUTINE DX(N.M.K.A.B.T.V.F)
C     REAL*8 A,B,T,V,F,U,U1,U2
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(10.10).B(10,10).F(10).V(10).U(10)
      DO 15 I=1,N
      F(I)=0.
      DO 15 J=1,N
15    F(I)=F(I)+A(I.J)*V(J)
      U(1)=U1(T)
      U(2)=U2(T)
      DO 25 I=1.N
      DO 25 J=1.M
25    F(I)=F(I)+B(I,J)*U(J)
      RETURN
      END
```

```
      FUNCTION SIMUL(N,A,X,EPS,INDIC,NRC)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 A,X,EPS,SIMUL
      DIMENSION IROW(50),JCOL(50),JORD(50),Y(50),A(NRC,NRC),X(N)
      MAX=N
      IF (INDIC.GE.0) MAX=N+1
      IF (N.LE.50) GO TO 5
      WRITE(6,200)
      SIMUL=0.
      RETURN
 5    DETER=1.
      DO 18 K=1,N
      KM1=K-1
      PIVOT=0.
      DO 11 I=1,N
      DO 11 J=1,N
      IF (K.EQ.1) GO TO 9
      DO 8 ISCAN=1,KM1
      DO 8 JSCAN=1,KM1
      IF (I.EQ.IROW(ISCAN)) GO TO 11
      IF (J.EQ.JCOL(JSCAN)) GO TO 11
 8    CONTINUE
 9    IF(DABS(A(I,J)).LE.DABS(PIVOT)) GO TO 11
      PIVOT=A(I,J)
      IROW(K)=I
      JCOL(K)=J
 11   CONTINUE
      IF(DABS(PIVOT).GT.EPS) GO TO 13
      SIMUL=0.
      RETURN
 13   IROWK=IROW(K)
      JCOLK=JCOL(K)
      DETER=DETER*PIVOT
      DO 14 J=1,MAX
 14   A(IROWK,J)=A(IROWK,J)/PIVOT
      A(IROWK,JCOLK)=1./PIVOT
      DO 18 I=1,N
      AIJCK=A(I,JCOLK)
      IF (I.EQ.IROWK) GO TO 18
      A(I,JCOLK)=-AIJCK/PIVOT
      DO 17 J=1,MAX
 17   IF(J.NE.JCOLK) A(I,J)=A(I,J)-AIJCK*A(IROWK,J)
 18   CONTINUE
      DO 20 I=1,N
      IROWI=IROW(I)
      JCOLI=JCOL(I)
      JORD(IROWI)=JCOLI
 20   IF(INDIC.GE.0) X(JCOLI)=A(IROWI,MAX)
      INTCH=0
      NM1=N-1
      DO 22 I=1,NM1
      IP1=I+1
      DO 22 J=IP1,N
```

```
      IF (JORD(J).GE.JORD(I)) GO TO 22
      JTEMP=JORD(J)
      JORD(J)=JORD(I)
      JORD(I)=JTEMP
      INTCH=INTCH+1
22    CONTINUE
      IF (INTCH/2*2.NE.INTCH) DETER=-DETER
      IF (INDIC.LE.0) GO TO 26
      SIMUL=DETER
      RETURN
26    DO 28 J=1,N
      DO 27 I=1,N
      IROWI=IROW(I)
      JCOLI=JCOL(I)
27    Y(JCOLI)=A(IROWI,J)
      DO 28 I=1,N
28    A(I.J)=Y(I)
      DO 30 I=1,N
      DO 29 J=1,N
      IROWJ=IROW(J)
      JCOLJ=JCOL(J)
29    Y(IROWJ)=A(I,JCOLJ)
      DO 30 J=1,N
30    A(I,J)=Y(J)
      SIMUL=DETER
      RETURN
200   FORMAT(2X,'TOO LONG')
      END
```

APPENDIX 2

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                           PROGRAM B                      C
C  THIS PROGRAM IS USED TO IDENTIFY UNKNOWN ORDER          C
C  MIMO LINEAR CONTINUOUS SYSTEMS.                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION YY(2,500),UU(2,500)
      INTEGER PP
      DIMENSION A(10,10),C(10,10),B(10,10),X(10)
555   FORMAT(8F9.4)
3     FORMAT(8F15.7)
1000  FORMAT(/10X,'SAMPLING START TIME = ',F10.6,10X,'SAMPLING END TIME
     *= ',F10.6,/11X,'SAMPLING PERIOD = ',F10.6)
1001  FORMAT(/10X,' THE A MATRIX ',/)
1002  FORMAT(9(1PE14.6))
1003  FORMAT(/10X,' INITIAL STATE VECTOR X: ',/)
1004  FORMAT(/10X,' THE B MATRIX ',/)
1006  FORMAT(/10X,' GAIN = ',E20.8)
1007  FORMAT(/10X,' THE C MATRIX ',/)
1008  FORMAT(8A2)
C AK REPRESENT SYSTEM FEEDBACK MATRIX.
C GAIN REPRESENT SYSTEM GAIN OF INPUT FUNCTION.
C X REPRESENT SYSTEM INITIAL CONDITION.
C U REPRESENT INPUT FUNCTION DATA.
      WRITE(6,100)
100   FORMAT(1H1)
      READ(5,1)N,MM,PP
1     FORMAT(3I2)
C     N=7
C     MM=1
C     PP=1
      WRITE(6,3340)N,MM,PP
3340  FORMAT(/10X,'N = ',I3,5X,'M = ',I3,5X,'P = ',I3//)
      WRITE(6,1001)
C     DO 1515 I=1,N
C     DO 1515 J=1,N
C     A(I,J)=0.
C     IF (J-I.EQ.1) A(I,J)=1.
C515  CONTINUE
C     A(7,1)=-281250.
C     A(7,2)=-3310875.
C     A(7,3)=-2814271.
C     A(7,4)=-8537039.
C     A(7,5)=-70342.
C     A(7,6)=-4097.
C     A(7,7)=-83.64
      DO 2 I=1,N
      READ(5,555)(A(I,J),J=1,N)
      WRITE(6,1002)(A(I,J),J=1,N)
2     CONTINUE
      WRITE(6,1004)
C     DO 1516 I=1,7
C516  B(I,1)=0.
```

```
C      B(6,1)=375000.
C      B(7,1)=-31333752.
       DO 99 I=1,N
       READ(5,555)(B(I,J),J=1,M4)
99     WRITE(6,1002)(B(I,J),J=1,M4)
       DO 30 I=1,PP
30     READ(5,555)(C(I,J),J=1,N)
C      DO 30 J=1,7
C0     C(I,J)=0.
C      C(1,1)=1.
       WRITE(6,1007)
       DO 4 I=1,PP
4      WRITE(6,1002)(C(I,J),J=1,N)
C      DO 98 I=1,N
       READ(5,555)(X(I),I=1,N)
C89    FORMAT(F9.5)
       WRITE(6,1003)
       WRITE(6,1002)(X(I),I=1,N)
C TZERO IS OUTPUT RESPONSE'S BEGINNING TIME.
C TF IS OUTPUT RESPONSE'S END TIME.
C DT IS SAMPLING PERIOD.
C IFQ IS HOW MANY TIMES WANT TO DISPLAY.
       READ(5,555)TZERO,TF,DT,FREQ,SCA
       IFQ=FREQ
       WRITE(6,1000)TZERO,TF,DT
       CALL INIT(YY,UU,N,MM,PP,A,B,C,TZERO,TF,DT,X)
       CALL BEGIN(YY,UU,MM,PP,N,DT)
       STOP
       END
```

```
      SUBROUTINE BEGIN(YY,UU,MM,PP,ITNN,SAT)
C UU IS SAMPLED INPUT DATA BUFFER. YY IS SAMPLED OUTPUT DATA BUFFER.
C HH IS INTEGRATING OUTPUT MARTIX THAT IS P MATRIX.
C VN IS INTEGRATING INPUT MARTIX THAT IS P MATRIX.
C TK IS SAMPLING TIME MATRIX.
C HNM IS DIFFERENTIALED MARTIX OF HN.
C BLWK IS LWK MATRIX THAT IS INCLUDEING L(T),H(T) AND K(T) MATRIX.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION YY(2,500),UU(2,500),S(48)
      INTEGER PP
      DIMENSION HN(36,48),VN(36,48),BUF(20,500)
     *,TK(36,48),HNM(36,48),BLWK(48,48),ALWK(48,48)
C NH IS ORDER. MM IS INPUTS. PP IS OUTPUT.
C ITNN IS TRUE SYSTEM ORDER.
C MM IS NUMBER OF INPUT. PP IS NUMBER OF OUTPUT
C SAT IS TRUE SAMPLING PERIOD.
      IYES=0
      IYEY=0
C IF A MATRIX IS CANONICAL FORM THEN IYES=IYEY=1
      INUM=10
C INUM IS TRUE SAMPLED PERIOD THAT SAMPLE ASSUMED SAMPLED PERIOD AGAIN.
C     TIN=0.0
C     DO 10 J=1,120
C     WRITE(6,1010)TIN,(YY(IY,J),IY=1,PP),(UU(IY,J),IY=1,MM)
C010  FORMAT(1X,F10.5,10E14.6)
C     TIN=TIN+0.005
C0    CONTINUE
      NN=3
C ************************************
C NN IS ASSUMED SYSTEM ORDER.
280   NNA=NN*(MM+2)
      NNMM=NN*MM
      NNPP=NN*PP
      NNAZ=NNA*INUM
C NNAZ IS TRUE NUMBERS OF SAMPLED DATA.
1000  FORMAT(3I2,2D14.6)
1011  FORMAT(F10.5,10D24.13)
C1010 FORMAT(1X,10(1PE14.6))
      DO 22 I= 1,NNMM
      DO 22 J=1,NNAZ
22    BUF(I,J)=0.0
C NOW IT DO FIRST INTEGRATION FOR INPUT SAMPLED DATA.
      DO 25 IA=1,MM
      IAA=NNMM-MM+IA
      BUF(IAA,1)=UU(IA,1)*SAT
      DO 25 II=2,NNAZ
      III=II-1
25    BUF(IAA,II)=BUF(IAA,III)+UU(IA,II)*SAT
C NOW DO SECOND,THIRD.... INTEGRATION FOR INPUT SAMPLED DATA.
      DO 26 IC=2,NN
      ID=IC-1
      DO 26 IF=1,MM
      ICC=NNMM-IC*MM+IF
```

```
      IDD=NNMM-ID*M4+IF
      BUF(ICC,1)=0.0
      BUF(ICC,1)=BUF(IDD,1)*SAT
      DO 25 IE=2,NNAZ
      IEEE=IE-1
25    BUF(ICC,IE)=BUF(ICC,IEEE)+BUF(IDD,IE)*SAT
C TRANSFER TRUE SAMPLED DATA TO ASSUMED SAMPLED DATA MATRIX.
C THAT'S ASSEMED SAMPLED PERIOD HAVE INUM TRUE SAMPLINGPERIOD.
      DO305 I= 1,NNMM
      DO 305 J= 1,NNA
      JBU=INUM+((J-1)*INUM)
      VN(I,J)=BUF(I,JBU)
305   CONTINUE
C DISPLAY THE VN MATRIX FOR TESTING.
      WRITE(6,1030)NNA,MM
      DO 121 IH=1,NNA
      WRITE(6,1020)(VN(I,IH),I=1,NNMM)
121   CONTINUE
1030  FORMAT(1H0,' COMPUTED INTEGRAL U(T) : NA*(M+2) =',I3,' ,M = ',I3)
      DO 15 I=1,NNPP
C NOW CALCULATE THE HN MATRIX.
      DO 15 J=1,NNA
15    HN(I,J)=0.0
      DO 303 I=1,NNPP
      DO 303 J= 1, NNAZ
303   BUF(I,J)=0.0
C NOW DO FIRST INTEGRATION OF OUTPUT SAMPLED DATA.
      DO 16 IA=1,PP
      IAA=NNPP-PP+IA
      BUF(IAA,1)=YY(IA,1)*SAT
      DO 16 II=2,NNAZ
      III=II-1
      BUF(IAA,II)=BUF(IAA,III)+YY(IA,II)*SAT
16    CONTINUE
C DO SECOND,THIRD..... INTEGRATION OF OUTPUT SAMPLED DATA.
      DO 17 IC=2,NN
      ID=IC-1
      DO 17IF=1,PP
      ICC=NNPP-IC*PP+IF
      IDD=NNPP-ID*PP+IF
      BUF(ICC,1)=BUF(IDD,1)*SAT
      DO 17 IE=2,NNAZ
      IEEE=IE-1
17    BUF(ICC,IE)=BUF(ICC,IEEE)+BUF(IDD,IE)*SAT
C TRANSFER BUFFER MATRIX TO ASSUMED SAMPLED PERIOD MATRIX.
      DO 304 I=1,NNPP
      DO 304 J=1,NNA
      JBU= INUM+((J-1)*INUM)
      HN(I,J)=BUF(I,JBU)
304   CONTINUE
C DISPLAY HN MATRIX FOR TESTING.
      WRITE(6,1040)NNA,PP
1040  FORMAT(1H0,' COMPUTED INTEGRAL Y(T) : NA*(M+2) =',I3,' , P = ',I3)
      DO 122 IH=1,NNA
      WRITE(6,1020)(HN(I,IH),I=1,NNPP)
122   CONTINUE
```

```
      DO 28 I=1,NNA
      DO 28 J=1,NN
28    TK(J,I)=1.0
C NOW CALCULATION THE K(T) MATRIX.
      JNNN=NN-1
      NNA1=NNA-1
      DO 30 ITT=1,NNA
30    TK(NN,ITT)=1.0
      DO 35 JI=1,NNA
C T IS ASSUMED SAMPLING PERIOD.
C T IS INUM TIMES OF TRUE SAMPLING PERIOD.
      T=JI*DAT*INUM
      DO 35 JH=1,JNNN
      ITT=NN-JH
      ITT1=ITT+1
35    TK(ITT,JI)=TK(ITT1,JI)*T/(JH*1.0)
C DISPLAY K(T) MATRIX FOR TESTING.
      WRITE(6,1060)NNA
1060  FORMAT(1H0,' COMPUTED K(T) : NA*(M+2) = ',I3)
      DO 123 IH=1,NNA
      WRITE(6,1020)(TK(I,IH),I=1,NN)
123   CONTINUE
34    CONTINUE
      WRITE(6,3278)
3278  FORMAT(1H1)
      NAL2=NA-1
C  INITIATE HN MATRIX AND PUT RESULTS TO HNM MATRIX
      DO 40 IQ=1,NAL2
      DO 40 IS=1,PP
      IS1 =(IQ-1)*PP+IS
      IS2 = IQ*PP+IS
      DO 40 IP = 1,NNA
40    HNM(IS1,IP)=HN(IS2,IP)
      DO 140 I=1,PP
      IY=NA*PP-PP+I
      DO 140 I2=1,NNA
C YY IS TRUE SAMPLE DATA. HNM IS ASSUMED SAMPLED DATA.
      JBB=INUM+((I2-1)*INUM)
      HNM(IY,I2)=YY(I,JBB)
140   CONTINUE
C DISPLAY HNM MATRIX FOR TESTING.
      WRITE(6,1070)NNA
1070  FORMAT(1H0,' COMPUTED HY(T) : NA(M+2) =',I3)
      DO 125 IH=1,NNA
      WRITE(6,1020) (HNM(I,IH),I=1,NN)
125   CONTINUE
C   COLLECT L(T),H(T),AND K(T) MATRIX TO BEAR MATRIX.
C THE DIMENSION OF BEAR MATRIX IS NN(M+2)*NN(M+2).
      WRITE(6,3278)
      DO 47 I=1,NNA
      DO 45 I1=1,NN
45    BEAR(I1,I)=HN(I1,I)
      IIM=NI*NN
      DO 46 I2=1,NIMM
      I1=I2+IIM
46    BEAR(I1,I)=VN(I2,I)
```

```
      DO 47 I3=1,NN
      I1=NN+NN*M+I3
47    BLWK(I1,I)=TK(I3,I)
      WRITE(6,198)NNA
198   FORMAT('0',' COMPUTED I MATRIX : NA(M+2) = ',I3)
      DO 199 IH=1,NNA
      WRITE(6,1020) (BLWK(I,IH),I=1,NNA)
199   CONTINUE
C NOW CALCULATE THE INVERSION OF BLWK MATRIX AND PUT RESULTS
C TO ALWK MATRIX. THE ORDER OF BLWK IS AAN(=NN(M+2)).
C NOW CALCULATE THE ASF MATRIX. THE RESULTS PUT ON BLWK MATRIX.
      DO 3338 IX=1,NNA
      DO 3338 IY=1,NNA
3338  ALWK(IX,IY)=BLWK(IX,IY)
      EPS=1.0E-19
      INDIC=-1
      DET=SIMUL(NNA,ALWK,3,EPS,INDIC,48)
48    DO 261 IX=1,NNA
      DO 261 IY=1,NNA
      BLWK(IX,IY)=0.0
      DO 261 IZ=1,NNA
      BLWK(IX,IY)=BLWK(IX,IY)+[illegible](IX,IZ)*ALWK(IZ,IY)
261   CONTINUE
C NOW BLWK MATRIX IS ASF MATRIX.
      WRITE(6,1100)
1100  FORMAT('1','COMPUTED INVERSE I MATRIX : ')
C DISPLAY INVERSION MATRIX FOR TESTING.
      DO 129 IH=1,NNA
      WRITE(6,1020) (ALWK(I,IH),I=1,NNA)
129   CONTINUE
1102  FORMAT(1H0,20[illegible])
      WRITE(6,[illegible])
      WRITE(6,2010)
      WRITE(6,2010)
      WRITE(6,2005) [illegible]
2005  FORMAT(/3X,'ACTUAL SYSTEM ORDER N=',I2)
      WRITE(6,2006) NA
2006  FORMAT(/3X,'ASSUMED SYSTEM ORDER NA=',I2)
      [illegible]*(1.0*[illegible])
      WRITE(6,2007) [illegible]
2007  FORMAT(/3X,'SAMPLING PERIOD T=',1PE9.2,////)
      WRITE(6,2008)
2008  FORMAT(3X,'THE ARGUMENT MATRIX IS :')
      DO 70 J=1,NNA
70    WRITE(6,1020) (BLWK(I,J),I=1,NNA)
C NOW BLWK [illegible] MATRIX
      DO 201 I=1,[illegible]
C NOW [illegible] MATRIX.
      DO 201 J=1,[illegible]
      A(I,J)=BLWK(I,J)
201   CONTINUE
      WRITE(6,2010)
2010  FORMAT(/,50('*'),/)
C DISPLAY A MATRIX FOR TESTING.
      WRITE(6,1302)
      DO 202 I=1,[illegible]
```

```
      WRITE(6,1020) (HN(I,J),J=1,NN)
202   CONTINUE
1302  FORMAT(5X,'ESTIMATED A MATRIX :')
C LET TK MATRIX EQUAL A MATRIX IN ORDER TO MAKE LOOP TO CALCULATE
C A*A ,A*A*A,..... ETC.
      DO 196 I=1,NN
      DO 196 J=1,NN
      TK(I,J)=HN(I,J)
196   CONTINUE
C DEFINE VN MATRIX TO REPRESENT B MATRIX.
      DO 203 III=1,MM
      NN1N=NN+III
      DO 203 I=1,NN
      VN(I,III)=BLWK(I,NN1N)
203   CONTINUE
C NOW MAKE LOOP FOR HNM REPRESENT R0,R1,.....,ETC.
      NN2N=(NN-1)*MM
      DO 204 ICC=1,NN2N,MM
      IID=NN+MM+ICC-1
      DO 205 IIE=1,MM
      IIF=IID+IIE
      DO 205 IIG=1,NN
      HNM(IIG,IIE)=BLWK(IIG,IIF)
205   CONTINUE
C NOW CALCULATE A*R1 AND PUT ON ALWK MATRIX.
      DO 262 IX=1,NN
      DO 262 IY=1,MM
      ALWK(IX,IY)=0.0
      DO 262 IZ=1,NN
      ALWK(IX,IY)=ALWK(IX,IY)+TK(IX,IZ)*HNM(IZ,IY)
262   CONTINUE
      DO 206 I=1,NN
      DO 206 J=1,MM
      VN(I,J)=VN(I,J)+ALWK(I,J)
206   CONTINUE
      DO 263 IX=1,NN
      DO 263 IY=1,NN
      ALWK(IX,IY)=0.0
      DO 263 IZ=1,NN
      ALWK(IX,IY)=ALWK(IX,IY)+HN(IX,IZ)*TK(IZ,IY)
263   CONTINUE
C MAKE LOOP TK REPRESENT A MAYTRIX WHICH MULTIPLY ITSELF.
      DO 208 I=1,NN
      DO 208 J=1,NN
      TK(I,J)=ALWK(I,J)
208   CONTINUE
C MAKE LOOP SET B=R0+A8R1+A*A*R2+....
204   CONTINUE
      WRITE(6,2010)
      WRITE(6,1300)
      DO 209 I=1,NN
      WRITE(6,1020) (VN(I,J),J=1,MM)
209   CONTINUE
C RESET TK MATRIX EQUAL A MATRIX.
      DO 211 I=1,NN
      DO 211 J=1,NN
```

```
      TK(I,J)=HN(I,J)
211   CONTINUE
C LET VN MATRIX EQUAL INITIAL XO MATRIX
      DO 212 I=1,NN
      I3I=NN*(MM+1)+1
      VN(I,1)=BLWK(I,I3I)
212   CONTINUE
C LET HNM MATRIX EQUAL G0,G1,,ETC
       DO 213 ICC=1,JMIN
       IDD=MM*(MM+1)+1+ICC
       DO 214 IIF=1,NN
       HNM(IIF,1)=BLWK(IIF,IDD)
214    CONTINUE
C LET TK MATRIX EQUAL A*G1,A*A*G2
       DO 264 IX=1,NN
       DO 264 IY=1,1
       ALWK(IX,IY)=0.0
       DO 264 IZ=1,NN
       ALWK(IX,IY)=ALWK(IX,IY)+TK(IX,IZ)*HNM(IZ,IY)
264    CONTINUE
       DO 215 I=1,NN
       VN(I,1)=VN(I,1)+ALWK(I,1)
215    CONTINUE
C MAKE LOOP TO CALCULATE A*A,A*A*A,,ETC
       DO 265 IX=1,NN
       DO 265 IY=1,NN
       ALWK(IX,IY)=0.0
       DO 265 IZ=1,NN
       ALWK(IX,IY)=ALWK(IX,IY)+HN(IX,IZ)*TK(IZ,IY)
265    CONTINUE
       DO 216 I=1,NN
       DO 216 J=1,NN
       TK(I,J)=ALWK(I,J)
216    CONTINUE
213    CONTINUE
       WRITE(6,2010)
       WRITE(6,1301)
       WRITE(6,1020)(VN(I,1),I=1,NN)
       WRITE(6,2010)
1301   FORMAT(5X,'ESTIMATED INITIAL STATE X VECTOR :')
1402   FORMAT(20(1PD13.11,','))
1300   FORMAT(5X,'ESTIMATED  B MATRIX :')
1020   FORMAT(9(1PE14.6))
C TEST0 IS TESTING MATRIX ELEMENT THAT HAS 0 ELEMENT
C TEST1 IS TESTING MATRIX ELEMENT THAT HAS 1 ELEMENT
       WRITE(6,3278)
       TEST0=1.0E-4
       TEST1=0.0001
       IMNI=NN-PP
       IYES=1
C THE IYES=1 REPRESENT A MATRIX IS CANONICAL FORM
C IYES=0 THEN A MATRIX IS NOT CANONICAL FORM
C THE FOLLOWING STATEMENTS ARE TESTING CANONICAL FORM
       DO 340 I=1,IMNI
       DO 340 J=1,PP
       TEB1=DABS(HN(I,J))
```

```
      IF(TEB1.GT.TESTO)IYES=0
 340  CONTINUE
      IF(IYES.EQ.0)GOTO359
      DO 341 I=1,INNI
      DO 341 J=1,INNI
      IN1=PP+J
      IF(I.NE.J)GOTO342
      TEB2=1.0-DABS(HN(I,IN1))
      IF(DABS(TEB2).GT.TEST1)GOTO345
      GOTO343
 342  TEB1=DABS(HN(I,IN1))
      IF(TEB1.GT.TESTO)GOTO345
 343  IYES=1
      IYEY=1
      GOTO341
 345  IYES=0
 341  CONTINUE
 355  WRITE(6,3010)IYES,IYEY
 359  IUM=0
 3010 FORMAT(/3X,'IYES=',I2,'  IYEY=',I2)
C IF IYES=0 AND IYEY=1 THEN TRUE SYSTEM ORDER IS NN-1
C IF NOT THEN GOTO BEGINNING
 357  IF(IYEY.NE.1)GOTO350
      IF(IYES.EQ.0)GOTO351
      IF(IYES.EQ.1)GOTO350
 352  IYEY=0
 350  NN=NN+1
      IF (NN.EQ.4)RETURN
      GOTO250
 351  NN=NN-1
      WRITE(6,2010)
      WRITE(4,2010)
      WRITE(6,3000)NN
 3000 FORMAT(/3X,'*****',' THE TRUE SYSTEM ORDER N=',I2)
 101  RETURN
      END
```

```
      SUBROUTINE INIT(Y.UU.N.M.P.A.B.C,X0,XL,H,V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(2,500),UU(2,500)
      INTEGER P
      DIMENSION A(10,10),B(10,10),C(10,10),F(10),V(10),TB(10,10)
      DIMENSION PA(10,10)
      CALL STMST(N,H,A,PA)
      CALL QT(N,M,A,PA,B,TB)
      DO 1007 I=1,P
      Y(I,1)=0.
      DO 1007 K=1,N
1007  Y(I.1)=Y(I.1)+C(I,K)*V(K)
      JJ=(XL-X0)/H+1.
      DO 134 K=2,JJ
      HK=(K-1)*H
      CALL DX(N,M,K,PA,TB,HK,V,F)
      DO 1006 I=1,P
      Y(I,K)=0.
      DO 1006 KI=1,N
1006  Y(I,K)=Y(I,K)+C(I,KI)*F(KI)
      DO 25 II=1,N
25    V(II)=F(II)
134   CONTINUE
      DO 1000 I=1,JJ
      XI=H*(I-1)
      UU(1,I)=U1(XI)
1000  UU(2,I)=U2(XI)
      WRITE(6,3278)M,P
3278  FORMAT(1H1,25X,'TIME    U(X), I= ',I3,10X,'Y(I), P= ',I3)
      DO 88 J=1,JJ
      XI=(J-1)*H
88    WRITE(6,89)XI,(UU(I,J),I=1,M),(Y(I,J),I=1,P)
89    FORMAT(7X,10E12.5)
      WRITE(6,3330)
3330  FORMAT(1H1)
      RETURN
      END
```

```
      FUNCTION U1(X)
      REAL*8 U1,X
      U1=1.+DSIN(2.0*X)
      RETURN
      END
```

```
      FUNCTION U2(X)
      REAL*8 U2,X
      U2=1.2+DCOS(2.4*X)
      RETURN
      END
```

```
      SUBROUTINE [illegible](N,M,A,PA,B,TB)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AA(10,10),A(10,10),PA(10,10),B(10,10),P(10,10)
      DIMENSION X(10),PB(10,10),TB(10,10)
      EPS=1.0E-15
      INDIC=-1
      DO 5 I=1,N
      DO 5 J=1,N
5     AA(I,J)=A(I,J)
      DET=SIMUL(N,AA,X,EPS,INDIC,10)
      DO 10 I=1,N
      DO 10 J=1,N
      P(I,J)=PA(I,J)
      IF (I.EQ.J) P(I,J)=P(I,J)-1.
10    CONTINUE
      DO 20 I=1,N
      DO 20 J=1,N
      PB(I,J)=0.
      DO 20 K=1,N
20    PB(I,J)=PB(I,J)+AA(I,K)*P(K,J)
      DO 30 I=1,N
      DO 30 J=1,N
      TB(I,J)=0.
      DO 30 K=1,N
30    TB(I,J)=TB(I,J)+PB(I,K)*B(K,J)
      RETURN
      END
```

```
      SUBROUTINE STMST(N,H,A,PH)
C     REAL*8 A,PH,AT,SAT,AC,H,R
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(10,10),PH(10,10),AT(10,10),SAT(10,10),AC(10)
      DO 43 I=1,N
      DO 43 J=1,N
43    PH(I,J)=0.
      DO 44 I=1,N
44    PH(I,I)=1.
      DO 46 I=1,N
      DO 46 J=1,N
      AT(I,J)=A(I,J)*H
46    SAT(I,J)=AT(I,J)
      R=2.
      DO 47 NK=1,15
      DO 48 I=1,N
      DO 48 J=1,N
48    PH(I,J)=PH(I,J)+SAT(I,J)
      DO 49 I=1,N
      DO 50 J=1,N
      AC(J)=0.
      DO 50 K=1,N
50    AC(J)=AC(J)+SAT(I,K)*AT(K,J)
      DO 49 J=1,N
49    SAT(I,J)=AC(J)/R
47    R=R+1.
      RETURN
      END


      SUBROUTINE DX(N,M,K,A,B,T,V,F)
C     REAL*8 A,B,T,V,F,U,U1,U2
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(10,10),B(10,10),F(10),V(10),U(10)
      DO 15 I=1,N
      F(I)=0.
      DO 15 J=1,N
15    F(I)=F(I)+A(I,J)*V(J)
      U(1)=U1(T)
      U(2)=U2(T)
      DO 25 I=1,N
      DO 25 J=1,M
25    F(I)=F(I)+B(I,J)*U(J)
      RETURN
      END
```

```
      FUNCTION SIMUL(N.A.X.EPS.INDIC.NRC)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 A,X,EPS,SIMUL
      DIMENSION IROW(50),JCOL(50).JORD(50).Y(50).A(NRC.NRC).X(N)
      MAX=N
      IF (INDIC.GE.0) MAX=N+1
      IF (N.LE.50) GO TO 5
      WRITE(6.200)
      SIMUL=0.
      RETURN
5     DETER=1.
      DO 18 K=1.N
      KM1=K-1
      PIVOT=0.
      DO 11 I=1,N
      DO 11 J=1,N
      IF (K.EQ.1) GO TO 9
      DO 8 ISCAN=1,KM1
      DO 8 JSCAN=1.KM1
      IF (I.EQ.IROW(ISCAN)) GO TO 11
      IF (J.EQ.JCOL(JSCAN)) GO TO 11
8     CONTINUE
9     IF(DABS(A(I.J)).LE.DABS(PIVOT)) GO TO 11
      PIVOT=A(I,J)
      IROW(K)=I
      JCOL(K)=J
11    CONTINUE
      IF(DABS(PIVOT).GT.EPS) GO TO 13
      SIMUL=0.
      RETURN
13    IROWK=IROW(K)
      JCOLK=JCOL(K)
      DETER=DETER*PIVOT
      DO 14 J=1.MAX
14    A(IROWK,J)=A(IROWK,J)/PIVOT
      A(IROWK,JCOLK)=1./PIVOT
      DO 18 I=1,N
      AIJCK=A(I.JCOLK)
      IF (I.EQ.IROWK) GO TO 18
      A(I,JCOLK)=-AIJCK/PIVOT
      DO 17 J=1,MAX
17    IF(J.NE.JCOLK) A(I,J)=A(I,J)-AIJCK*A(IROWK,J)
18    CONTINUE
      DO 20 I=1,N
      IROWI=IROW(I)
      JCOLI=JCOL(I)
      JORD(IROWI)=JCOLI
20    IF(INDIC.GE.0) X(JCOLI)=A(IROWI,MAX)
      INTCH=0
      NM1=N-1
      DO 22 I=1,NM1
      IP1=I+1
      DO 22 J=IP1,N
```

```
      IF (JORD(J).GE.JORD(I)) GO TO 22
      JTEMP=JORD(J)
      JORD(J)=JORD(I)
      JORD(I)=JTEMP
      INTCH=INTCH+1
22    CONTINUE
      IF (INTCH/2*2.NE.INTCH) DETER=-DETER
      IF (INDIC.LT.0) GO TO 26
      SIMUL=DETER
      RETURN
26    DO 28 J=1,N
      DO 27 I=1,N
      IROWI=IROW(I)
      JCOLI=JCOL(I)
27    Y(JCOLI)=A(IROWI,J)
      DO 28 I=1,N
28    A(I,J)=Y(I)
      DO 30 I=1,N
      DO 29 J=1,N
      IROWJ=IROW(J)
      JCOLJ=JCOL(J)
29    Y(IROWJ)=A(I,JCOLJ)
      DO 30 J=1,N
30    A(I,J)=Y(J)
      SIMUL=DETER
      RETURN
200   FORMAT(2X,'TOO LONG')
      END
```

APPENDIX 3

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                        PROGRAM C                  C
C THIS PROGRAM IS USED TO IDENTIFY UNKNOWN ORDER     C
C LINEAR DISCRETE-TIME SYSTEMS                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DIMENSION YY(150),UU(150),A(150,20),B(150,20),BB(20,20)
      DIMENSION CC(20,20),AA(20,20),EK(20),RR(20,20),BBI(20,20)
      DIMENSION SS(20,20),ALPHA(20),AK(4),BK(5),U(150),Z(150)
      DOUBLE PRECISION YY,UU,A,B,RR,SS,AA,BB,BBI,AK,BK,U,Z,JM,T
      DOUBLE PRECISION CC,EK,U0,Z0
      READ(5,999)N
999   FORMAT(I3)
      AK(1)=-0.0066
      AK(2)=0.4888
      AK(3)=-1.7512
      AK(4)=2.2605
      BK(1)=0.0066
      BK(2)=-0.1245
      BK(3)=0.0289
      BK(4)=0.0975
      BK(5)=0.
      Z0=0.
      U0=0.
      JM=0.
      DO 117 J=1,N
      JM=JM+1.
      U(J)=JM*DTAN(60*JM)
117   CONTINUE
      Z(1)=AK(4)*Z0+BK(4)*U0+BK(5)*U(1)
      Z(2)=AK(3)*Z0+AK(4)*Z(1)+BK(3)*U0+BK(4)*U(1)+BK(5)*U(2)
      Z(3)=AK(2)*Z0+AK(3)*Z(1)+AK(4)*Z(2)+BK(2)*U0+BK(3)*U(1)+BK(4)*
     *U(2)+BK(5)*U(3)
      Z(4)=AK(1)*Z0+AK(2)*Z(1)+AK(3)*Z(2)+AK(4)*Z(3)+BK(1)*U0+BK(2)*
     *U(1)+BK(3)*U(2)+BK(4)*U(3)+BK(5)*U(4)
      DO 118 K=5,N
      Z(K)=AK(1)*Z(K-4)+AK(2)*Z(K-3)+AK(3)*Z(K-2)+AK(4)*Z(K-1)
     *+BK(1)*U(K-4)+BK(2)*U(K-3)+BK(3)*U(K-2)+BK(4)*U(K-1)+BK(5)*U(K)
118   CONTINUE
      UU(1)=U0
      YY(1)=Z0
      DO 119 I=1,N
      UU(I+1)=U(I)
      YY(I+1)=Z(I)
119   CONTINUE
      WRITE(6,33)N
33    FORMAT(//5X,'THE SAMPING LENGTH=',I3)
      CALL ORD(N,YY,UU,M,A,B)
      NII=N-M+1
      NJJ=(M+1)*2
      NJP=2*M+1
      M1=M+1
      IF(M.GT.5)GO TO 100
      WRITE(6,111)M
111   FORMAT(//5X,'THE SYSTEM ORDER=',I2)
      DO 2 II=1,NJP
      DO 2 JJ=1,NJJ
      CC(II,JJ)=0.
2     CONTINUE
```

```
      DO 3 II=1,M
      CC(II.II)=1.
3     CONTINUE
      DO 5 LI=MM,NJP
      LL=LI+1
      CC(LI.LL)=1.
5     CONTINUE
      DO 7 IA=1,NJJ
      EK(IA)=0.
7     CONTINUE
      EK(MM)=1.
      CALL TRPRA(A,NII,NJJ,AA)
      WRITE(6,15)N
15    FORMAT(//5X,' THE SAMPING LENGTH=',I3)
      WRITE(6,81)NJJ,NJJ,(I,I=1,6)
81    FORMAT(//' MATRIX AA(',I2,',',I2,')='/5X,6(14X,I1))
      DO 981 I=1,NJJ
981   WRITE(6,990)I,(AA(I,J),J=1,NJJ)
990   FORMAT('0 ROW',I3,4X,6(1PD15.6)/(12X,6(1PD15.6)))
      CALL TRPRA(B,NII,NJP,BB)
      WRITE(6,80)NJP,NJP,(I,I=1,6)
80    FORMAT(//' MATRIX BB(',I2,',',I2,')='/5X,6(14X,I1))
      DO 98 I=1,NJP
98    WRITE(6,90)I,(BB(I,J),J=1,NJP)
90    FORMAT('0 ROW',I3,4X,6(1PD15.6)/(12X,6(1PD15.6)))
      CALL MATINV(BB,NJP,BBI,IERR)
      CALL MAMULT(BBI,CC,RR,NJP,NJP,NJJ)
      CALL MAMULT(RR,AA,SS,NJP,NJJ,NJJ)
      DO 777 I=1.NJP
      ALPHA(I)=0.
      DO 777 J=1,NJJ
      ALPHA(I)=ALPHA(I)+SS(I.J)*EK(J)
777   CONTINUE
      WRITE(6,35)
35    FORMAT(/3X,'THE PRINCIPAL ALPHA VECTOR='/)
      WRITE(6,20)(AK(I),I=1,4),(BK(I),I=1,5)
      WRITE(6,45)
45    FORMAT(/3X,'THE ALPHA VECTOR='/)
      WRITE(6,20)(ALPHA(I),I=1,NJP)
20    FORMAT(5(1PD15.6))
100   STOP
      END
```

```
      SUBROUTINE ORD(N,YY,UU,M,A,B)
      DIMENSION YY(150),UU(150),A(150,20),B(150,20),IDX(20)
      DOUBLE PRECISION YY,UU,A,B
      K=0
3     CALL COMB(N,K,YY,UU,A,B)
      NII=N-K+1
      NJJ=(K+1)*2
      NJP=NJJ-1
      EPS=1.0E-6
      WRITE(6,83)NII,NJJ,(I,I=1,6)
83    FORMAT(//' MATRIX A(',I2,',',I2,')='/5X,6(14X,I1))
      DO 984 I=1,NII
984   WRITE(6,994)I,(A(I,J),J=1,NJJ)
994   FORMAT('0 ROW',I3,4X,6(1PD15.6)/(12X,6(1PD15.6)))
      WRITE(6,86)NII,NJP,(I,I=1,6)
86    FORMAT(//' MATRIX B(',I2,',',I2,')='/5X,6(14X,I1))
      DO 87 I=1,NII
87    WRITE(6,44)I,(B(I,J),J=1,NJP)
44    FORMAT('0 ROW',I3,4X,6(1PD15.6)/(12X,6(1PD15.6)))
      CALL RANK(A,IDX,NII,NJJ,MRANKA,EPS)
      NII=N-K+1
      NJP=2*K+1
      EPS=1.0E-6
      CALL RANK(B,IDX,NII,NJP,MRANKB,EPS)
      IDIF=MRANKA-MRANKB
      WRITE(6,10)K,MRANKA,MRANKB,IDIF
10    FORMAT(//'ITER. NO.=',I2/' RANKA=',I2/' RANKB=',I2/' DIF=',I2/)
      IF(IDIF.EQ.0) GO TO 2
      K=K+1
      IF(K-5)3,3,5
2     M=K
5     RETURN
      END
```

```
      SUBROUTINE COMB(N.K.Y.U.A.B)
      DIMENSION A(150,20),B(150,20),Y(150),U(150)
      DOUBLE PRECISION A,B,Y,U
      K1=K+1
      NK1=N-K+1
      N1=N+1
      DO 1 J=1,K1
      IV=0
      J1=J-1
      NKJ1=NK1+J1
      MX=J1+1
      DO 1 II=J.NKJ1
      IV=IV+1
      A(IV,MX)=Y(II)
1     CONTINUE
      DO 2 J=1.K1
      IV=0
      J1=J-1
      NKJ1=NK1+J1
      MM=K+J1+2
      DO 2 II=J,NKJ1
      IV=IV+1
      A(IV.MM)=U(II)
2     CONTINUE
      IF(K.EQ.0) GO TO 3
      DO 4 II=1,NK1
      DO 4 MM=1.K
      B(II,MM)=A(II,MM)
4     CONTINUE
      K2=K+2
      NII=(K+1)*2
      DO 10 II=1,NK1
      DO 10 MM=K2,NII
      NN=MM-1
      B(II,NN)=A(II,MM)
10    CONTINUE
      RETURN
3     DO 11 II=1.N1
      B(II,1)=A(II,2)
11    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRPRA(A.MI.MJ.XTX)
      DIMENSION A(150,20),XTX(20,20)
      DOUBLE PRECISION A,XTX,TEMP
      DO 1 I=1,MJ
      DO 1 J=1,MJ
      TEMP=0.
      DO 2 L=1,MI
    2 TEMP=TEMP+A(L.I)*A(L.J)
      XTX(I,J)=TEMP
    1 CONTINUE
      RETURN
      END


      SUBROUTINE RANK(A.IDX.NII.NJJ,MRANK,EPS)
      DIMENSION IDX(20),B(150,20),A(150,20)
      DOUBLE PRECISION TEMP,D,B,A
      NI=NII
      NJ=NJJ
      IF(NI.GT.NJ) GO TO 3
      DO 1 I=1,NII
      DO 1 J=1,NJJ
      B(J,I)=A(I,J)
    1 CONTINUE
      NI=NJJ
      NJ=NII
    3 DO 2 I=1,NII
      DO 2 J=1,NJJ
      B(I,J)=A(I,J)
    2 CONTINUE
      MRANK=NJ
      DO 10 I=1,NJ
      IDX(I)=1
      WS1=DABS(B(I,I))
      KI=I
      DO 4 II=I,NI
      WS2=DABS(B(II.I))
      IF(WS1.GE.WS2) GO TO 4
      WS1=WS2
      KI=II
    4 CONTINUE
      IF(KI.EQ.I) GO TO 5
      DO 6 J=1,NJ
      TEMP=B(I,J)
      B(I,J)=B(KI,J)
      B(KI,J)=TEMP
    6 CONTINUE
    5 IF(DABS(B(I.I)).GT.EPS) GO TO 7
      MRANK=MRANK-1
      IDX(I)=0
      GO TO 10
```

```
7      D=B(I,I)
       I1=I+1
       DO 8 L=I1,NI
       TEMP=B(L,I)
       DO 8 J=I,NJ
       B(L,J)=B(L,J)-B(I,J)/D*TEMP
8      CONTINUE
10     CONTINUE
       RETURN
       END
```

```
      SUBROUTINE MATINV(A,KC,AI,IERR)
      DIMENSION A(20,20),B(20,20),AI(20,20)
      DOUBLE PRECISION A,B,AI,TEMP,COMP
      WRITE(6,20)
 20   FORMAT(/3X,'THE ORINGINAL MATRIX'/)
      WRITE(6,21)((A(I,J),J=1,KC),I=1,KC)
 21   FORMAT(5(1PD15.5))
      N=1
      IERR=1
      DO 1 I=1,KC
      DO 1 J=1,KC
      AI(I,J)=0.
      B(I,J)=A(I,J)
 1    CONTINUE
      DO 2 I=1,KC
      AI(I,I)=1.0
 2    CONTINUE
      DO 3 I=1,KC
      COMP=0.0
      K=I
 6    IF(DABS(B(K,I))-DABS(COMP))5,5,4
 4    COMP=B(K,I)
      N=K
 5    K=K+1
      IF(K-KC)6,6,7
 7    IF(B(N,I))8,51,8
 8    IF(N-I)51,12,9
 9    DO 10 M=1,KC
      TEMP=B(I,M)
      B(I,M)=B(N,M)
      B(N,M)=TEMP
      TEMP=AI(I,M)
      AI(I,M)=AI(N,M)
 10   AI(N,M)=TEMP
 12   CONTINUE
      TEMP=B(I,I)
      DO 13 M=1,KC
      AI(I,M)=AI(I,M)/TEMP
      B(I,M)=B(I,M)/TEMP
 13   CONTINUE
      DO 16 J=1,KC
      IF(J-I)14,16,14
 14   IF(B(J,I))15,16,15
 15   CONTINUE
      TEMP=B(J,I)
      DO 17 N=1,KC
      AI(J,N)=AI(J,N)-TEMP*AI(I,N)
 17   B(J,N)=B(J,N)-TEMP*B(I,N)
 16   CONTINUE
 3    CONTINUE
      WRITE(6,22)
 22   FORMAT(/3X,'THE INVERSE MATRIX'/)
      WRITE(6,21)((AI(I,J),J=1,KC),I=1,KC)
```

```
      RETURN
51    WRITE(6,52)
52    FORMAT(6X,22HTHE MATRIX IS SINGULAR)
      IERR=0
      RETURN
      END


      SUBROUTINE MAMULT (AD,BD,CD,L,M,N)
      DIMENSION AD(20,20),BD(20,20),CD(20,20)
      DOUBLE PRECISION AD,BD,CD
      DO 10 I=1,L
      DO 10 J=1,N
      CD(I,J)=0.
      DO 10 K=1,M
10    CD(I,J)=CD(I,J)+AD(I,K)*BD(K,J)
      RETURN
      END
```

APPENDIX 4

A4−1 Reduction of high order system

In practice, even when it is possible to identify the parameters of complex and high order systems, the analysis, optimization and adaptation would require a large amount of computation. One way of overcoming these computational difficulities is to use a low order model of the high order system which is computationally and analytically more tractable than the actual system, yet still provides sufficient information about the original system.

Let the given high order system $\bar{S}$ be described by

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t) \qquad (A4-1)$$

$$\underline{y}(t) = C\,\underline{x}(t) \qquad (A4-2)$$

where

$\underline{x}(t) \in R^{n\times 1}$ is a state vector

$\underline{y}(t) \in R^{p\times 1}$ is a output vector

$\underline{u}(t) \in R^{m\times 1}$ is a input vector

and

$A \in R^{n\times n}$, $B \in R^{n\times m}$ and $C \in R^{p\times n}$ are constant matrices which are either known or unknown.

Assuming that only the system inputs and outputs are accessible for measurement, our objective is to provide a low order model of the high order system which is computationally or analytically more tractable than the actual system, yet provides sufficient information about the system.

Let the low order model of order q, where $p \leq q \leq n$ be described by :

$$\dot{\underline{x}}_r(t) = A_r \underline{x}_r(t) + B_r \underline{u}(t) \qquad \text{(A4-3)}$$

$$\underline{y}_r(t) = C_r \underline{x}_r(t) \qquad \text{(A4-4)}$$

where

$\underline{x}_r(t) \in R^{q\times 1}$

$\underline{u}(t) \in R^{m\times 1}$

$\underline{y}_r(t) \in R^{p\times 1}$

$A_r \in R^{q\times q}$, $B_r \in R^{q\times m}$ and $C_r \in R^{p\times q}$ are unknown constant matrices to be determined, such that for the same input $\underline{u}(t)$, the output $\underline{y}_r(t)$ is sufficiently close to $\underline{y}(t)$.

Referring to equation (3-49), if $\underline{y}(t)$, $\underline{u}(t)$ and their q-successive integrals are measured at $q(m+2)$ successive samples of time with a sampling interval (T), then $\dot{R}(T)$ and V(T) can be constructed. Now if for some sampling interval T, and some input $\underline{u}(t)$, V(T) is non-singular then :

$$[A_r \ \ P_r \ \ M_r] = \dot{R}(T)[V(T)]^{-1} \qquad \text{(A4-5)}$$

where

$\dot{R}(T) \in R^{q\times q(m+2)}$, $V(T) \in R^{q(m+2)\times q(m+2)}$

Now the reduced model system matrix A_r is obtained from equation (3-67) while the input matrix B_r, the initial state

$\underline{x}_r(t_0)$ and the output matrix C_r are given by :

$$B_r = P_{r0} + A_r P_{r1} + \cdots\cdots + A_r^{q-1} P_{r(q-1)} \qquad \text{(A4-6)}$$

$$\underline{x}_r(t_0) = M_{r0} + A_r M_{r1} + \cdots\cdots + A_r^{q-1} M_{r(q-1)} \qquad \text{(A4-7)}$$

and

$$C_r = [\ I_{p\times p} \,\vdots\, O_{p\times(q-p)}\ ] \qquad \text{(A4-8)}$$

Since the order of the model (q) is less than the actual system order n, then for some input $\underline{u}(t)$, the reduced parameters A_r, B_r are not unique i.e., different models can be obtained for different sampling intervals.

So far, there is no specific method available to determine the sampling interval which provides us with the optimum low-order model with respect to a specified criterion. But since the response of the reduced-order system is required to be close to the response of the actual system during the transient and steady state intervals, the sampling interval is chosen such that the response is measured over both intervals.

A 4 — 2 Illustrative examples

To demonstrate the use of the above method to obtain a reduced order model, we consider the following examples. For the numerical computation the Program B in Appendix 2 was used.

To assess the extent of error introduced in the reduced model, the following equation is used to give an index of error between the response of the actual system and the reduced system.

$$S = \frac{1}{T_s}\int_0^{t_s} [\, y(t) - y_r(t)\,]^2 \, dt \qquad (A4\text{-}9)$$

where

$y(t)$ is the response of the actual system

$y_r(t)$ is the response of the reduced system

t_s is the time interval of the response during the transient and steady state

In deriving the reduced order system, the designer has the choice of specifying a time interval for the collection of the measurement data. In the absence of a particular method to compute optimum sampling interval T, we can search over T to find the reduced model for which we have a minimum value of S (A 4-9). The following equation gives a relationship between the sampling interval T and the settling time t_s of the system to be reduced.

$$T = K\, t_s \qquad (A4\text{-}2)$$

where K is a proportional constant

The sampling interval T was then varied to observe its effect on S.

(a) Example A4-1

Consider the following 3rd order system

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 1.000 \\ -15.000 & -23.000 & -9.000 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

$$+\begin{bmatrix} 0.000 \\ 0.000 \\ 1.000 \end{bmatrix} u(t)$$

$$C = [\ 1.000 \quad 0.000 \quad 0.000\ ]$$

where

$$u(t) = 1.0 + \sin(1.2t)$$

and the settling time $t_s = 5$ seconds

In this example, the sampling interal is varied from 0.024 seconds to 0.051 seconds to search for good approximations of the actual system.

The results for three values of the sampling intervals are shown below

(1) $T = 0.031$ seconds

$$A_r = \begin{bmatrix} -1.065814D-14 & 1.000000D+00 \\ -1.640349D+00 & -1.965433D+00 \end{bmatrix}$$

$$B_r = \begin{bmatrix} 3.360006D-03 \\ 9.907115D-02 \end{bmatrix}$$

(2) $T = 0.032$ seconds

$$A_r = \begin{bmatrix} 1.421085D-14 & 1.000000D+00 \\ -1.735320D+00 & -2.125264D+00 \end{bmatrix}$$

$$B_r = \begin{bmatrix} 1.938859D-04 \\ 1.116925D-01 \end{bmatrix}$$

(3) $T = 0.034$ seconds

$$A_r = \begin{bmatrix} 1.278977D-13 & 1.000000D+00 \\ -1.894064D+00 & -2.402988D+00 \end{bmatrix}$$

$$B_r = \begin{bmatrix} 2.155078D-04 \\ 1.356261D-01 \end{bmatrix}$$

The summaries of the errors of reduction order with reference to (A4-9) is tabulated in Table A4-1. this table supports the validity of the method used here. Three step responses of the original third order system and the reduced order system are shown in Figs. A4-1 ~ A4-3.

Table A4-1 Errors in order reduction

Sampling interval T = K t_s (seconds)	Error S	Standard deviation SD
0.051 (K = 0.0102)	0.22908104 D − 03	0.15026424 D − 01
0.049 (K = 0.0098)	0.64350131 D − 04	0.88089981 D − 02
0.048 (K = 0.0096)	0.18329792 D − 04	0.43059378 D − 02
0.047 (K = 0.0094)	0.78492465 D − 05	0.28147325 D − 02
0.046 (K = 0.0092)	0.39397185 D − 05	0.19929566 D − 02
0.045 (K = 0.0090)	0.21010523 D − 05	0.14548334 D − 02
0.044 (K = 0.0088)	0.11223660 D − 05	0.10630063 D − 02
0.043 (K = 0.0086)	0.57305597 D − 06	0.75939109 D − 03
0.042 (K = 0.0084)	0.27297749 D − 06	0.52401464 D − 03
0.041 (K = 0.0082)	0.13976599 D − 06	0.37488899 D − 03
0.040 (K = 0.0080)	0.13539503 D − 06	0.36891877 D − 03
0.039 (K = 0.0078)	0.24447464 D − 06	0.49565245 D − 03
0.038 (K = 0.0076)	0.46467705 D − 06	0.68323370 D − 03
0.037 (K = 0.0074)	0.80205901 D − 06	0.89749529 D − 03
0.036 (K = 0.0072)	0.22576261 D − 05	0.15069329 D − 02
0.035 (K = 0.0070)	0.30112665 D − 05	0.17405738 D − 02
0.034 (K = 0.0068)	0.40902945 D − 05	0.20289596 D − 02
0.032 (K = 0.0066)	0.82812593 D − 05	0.28887846 D − 02
0.031 (K = 0.0062)	0.12509553 D − 04	0.35523659 D − 02
0.030 (K = 0.0060)	0.19879831 D − 04	0.44817411 D − 02
0.029 (K = 0.0058)	0.33744068 D − 04	0.58462361 D − 02
0.028 (K = 0.0056)	0.62885326 D − 04	0.79972735 D − 02
0.027 (K = 0.0054)	0.32255553 D − 04	0.57034646 D − 02
0.026 (K = 0.0052)	0.87943100 D − 04	0.94481594 D − 02
0.025 (K = 0.0050)	0.47776769 D − 03	0.22224150 D − 01
0.024 (K = 0.0048)	0.54496945 D − 01	0.24989445 D + 00

(1) $T = 0.031$ seconds

$$A_r = \begin{bmatrix} -1.065814D-14 & 1.000000D+00 \\ -1.640349D+00 & -1.965433D+00 \end{bmatrix}$$

$$B_r = \begin{bmatrix} 3.360006D-03 \\ 9.907115D-02 \end{bmatrix}$$

$$S = 0.12509553D-04$$

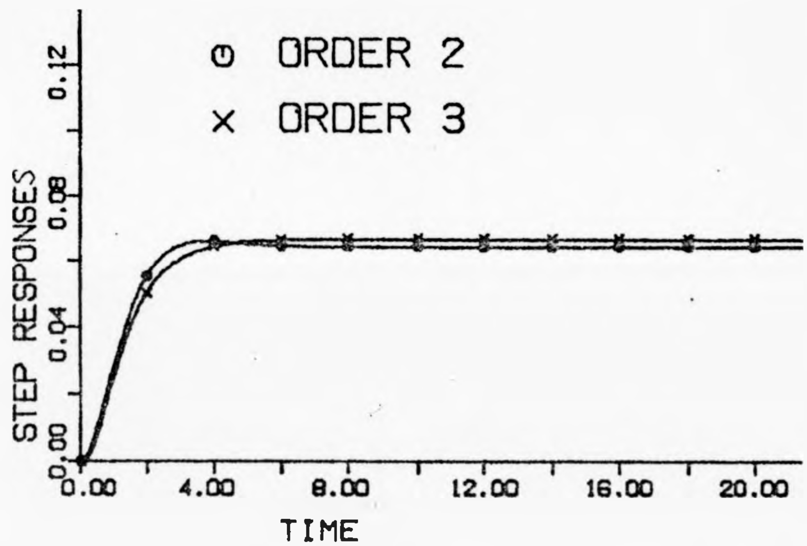

Fig. A4-1 The step responses of $y(t)$ and $y_r(t)$

(2) $T = 0.032$ seconds

$$A_r = \begin{bmatrix} 1.421085\,D-14 & 1.000000\,D+00 \\ -1.735320\,D+00 & -2.125264\,D+00 \end{bmatrix}$$

$$B_r = \begin{bmatrix} 1.938859\,D-04 \\ 1.116925\,D-01 \end{bmatrix}$$

$$S = 0.82812593\,D-05$$

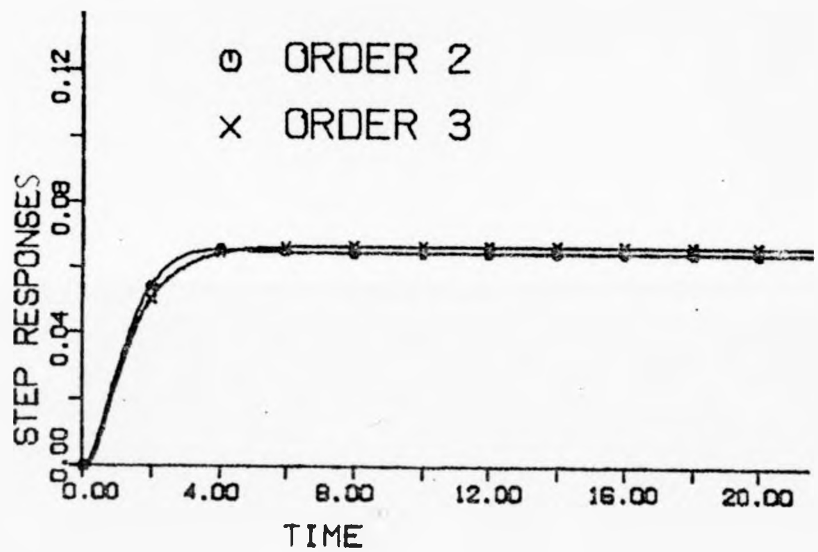

Fig. A 4−2 The step responses of $y(t)$ and $y_r(t)$

(3) $T = 0.034$ seconds

$$A_r = \begin{bmatrix} 1.278977\,D-13 & 1.000000\,D+00 \\ -1.894064\,D+00 & -2.402988\,D+00 \end{bmatrix}$$

$$B_r = \begin{bmatrix} 2.155078\,D-04 \\ 1.356261\,D-01 \end{bmatrix}$$

$$S = 0.40902945\,D-05$$

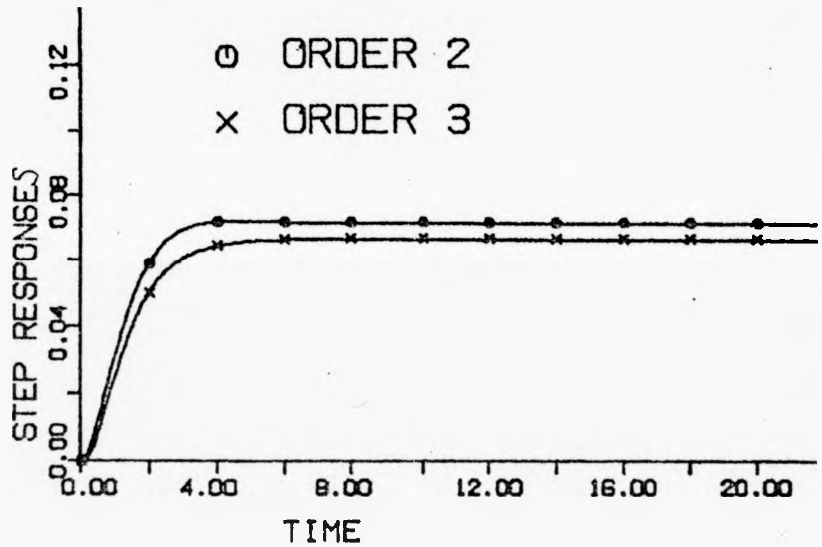

Fig. A 4−3 The step responses of $y(t)$ and $y_r(t)$

(b) Example A4−2

consider the following 4th order system

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0.000 & 1.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 1.000 \\ -150.000 & -245.000 & -113.000 & -19.000 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0.000 \\ 0.000 \\ 0.000 \\ 1.000 \end{bmatrix} u(t)$$

$$C = [\ 1.000 \quad 0.000 \quad 0.000 \quad 0.000\ ]$$

where

$$u(t) = 1.0 + \sin(1.2\,t)$$

and the settling time $t_s = 5$ seconds

In this example, the sampling interval is varied from 0.030 seconds (K = 0.0060) to 0.050 seconds (K = 0.0100) to search for the good approximations of the actual system. The results for two sampling intervals are shown below.

(1) T = 0.035 seconds

$$A_r = \begin{bmatrix} 1.598721\,D-13 & 1.000000\,D+00 \\ -1.334360\,D+00 & -1.192328\,D+00 \end{bmatrix}$$

$$B_r = \begin{bmatrix} 7.261047\,D-04 \\ 7.011215\,D-03 \end{bmatrix}$$

(2) T = 0.036 seconds

$$A_r = \begin{bmatrix} 1.527667\,D-13 & 1.000000\,D+00 \\ -1.432157\,D+00 & -1.418701\,D+00 \end{bmatrix}$$

A summary of the errors in order reduction with reference to (A4−9) is tabulated in Table A4−2 which supports the validity of the method used here. Two step responses of the original fourth order system and the reduced order system are shown in Figs. A 4−4 and A 4−5.

Table A 4−2 Errors in order reduction

Sampling interval $T = KT_S$ (second)	Error S	Standard deviation SD
0.030 (K = 0.0060)	0.12510514 D−00	0.33797768 D−00
0.031 (K = 0.0062)	0.41404879 D−04	0.64472895 D−02
0.032 (K = 0.0064)	0.34165453 D−05	0.18514778 D−02
0.033 (K = 0.0066)	0.15334797 D−05	0.12406044 D−02
0.034 (K = 0.0068)	0.89201300 D−06	0.94630327 D−03
0.035 (K = 0.0070)	0.56369886 D−06	0.75233096 D−03
0.036 (K = 0.0072)	0.36736633 D−06	0.60739234 D−03
0.037 (K = 0.0074)	0.24032483 D−06	0.49130227 D−03
0.038 (K = 0.0076)	0.15452157 D−06	0.39397669 D−03
0.039 (K = 0.0078)	0.95389287 D−07	0.30956356 D−03
0.040 (K = 0.0080)	0.54629657 D−07	0.23428070 D−03
0.041 (K = 0.0082)	0.27284472 D−07	0.16557742 D−03
0.042 (K = 0.0084)	0.10441184 D−07	0.10243271 D−03
0.043 (K = 0.0086)	0.27090508 D−08	0.52178658 D−04
0.044 (K = 0.0088)	0.42217353 D−08	0.65140551 D−04
0.045 (K = 0.0090)	0.17135131 D−07	0.13192978 D−03
0.046 (K = 0.0092)	0.48735173 D−07	0.22135025 D−03
0.047 (K = 0.0094)	0.11659609 D−06	0.34240405 D−03
0.048 (K = 0.0096)	0.27670013 D−06	0.52754500 D−03
0.049 (K = 0.0098)	0.76365707 D−06	0.87662356 D−03
0.050 (K = 0.0100)	0.32906138 D−03	0.17885979 D−01

(1) $T = 0.035$ seconds

$$A_r = \begin{bmatrix} 1.598721\,D-13 & 1.000000\,D+00 \\ -1.334360\,D+00 & -1.192328\,D+00 \end{bmatrix}$$

$$B_r = \begin{bmatrix} 7.261047\,D-04 \\ 7.011215\,D-03 \end{bmatrix}$$

$$S = 0.56369886\,D-06$$

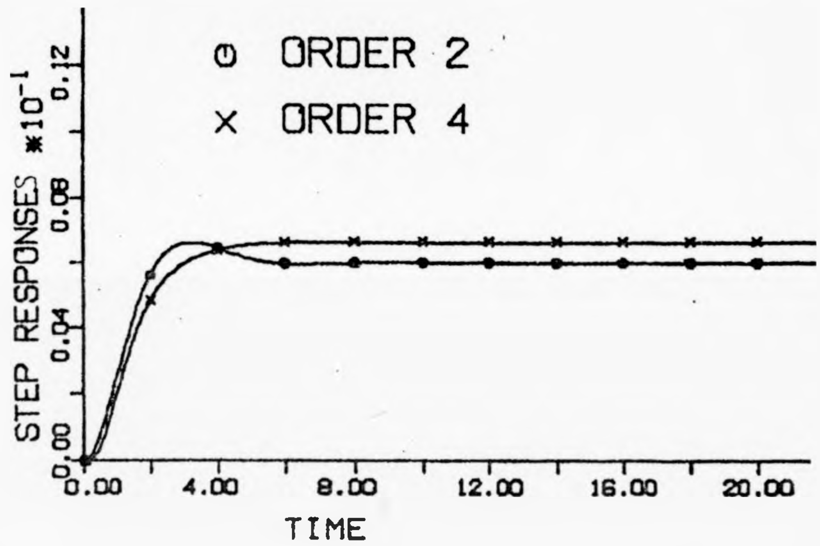

Fig. A 4—4 The step responses of $y(t)$ and $y_r(t)$

(2) $T = 0.036$ seconds

$$A_r = \begin{bmatrix} 1.527667\,D-13 & 1.000000\,D+00 \\ -1.432157\,D+00 & -1.418701\,D+00 \end{bmatrix}$$

$$B_r = \begin{bmatrix} 2.607955\,D-04 \\ 8.273771\,D-03 \end{bmatrix}$$

$$S = 0.36736633\,D-06$$

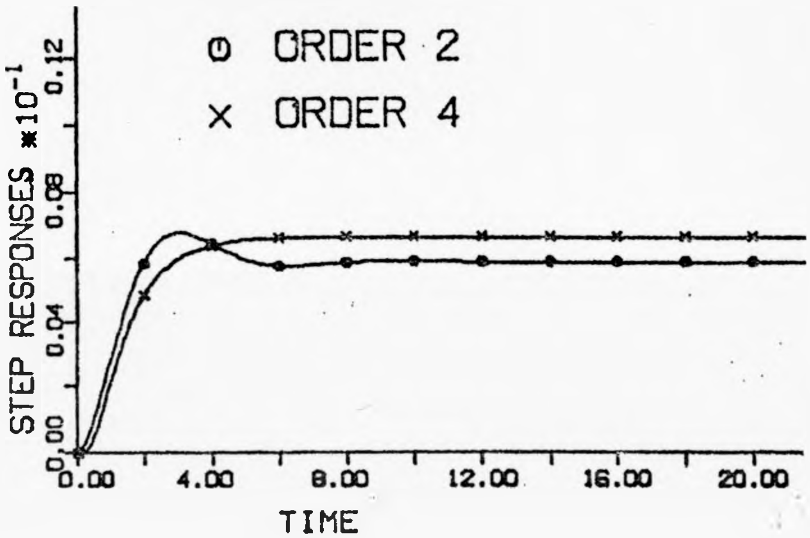

Fig. A 4−5 The step responses of $y(t)$ and $y_r(t)$

TITLE

IDENTIFICATION AND DESIGN
OF CONTROL SYSTEMS

by
CHENG-KUNG YU

AUTHOR

UNIVERSITY OF WARWICK

UNIVERSITY 1985